



3D Skeletons: A State-of-the-Art Report

Andrea Tagliasacchi, Thomas Delame, Michela Spagnuolo, Nina Amenta,
Alexandru C Telea

► To cite this version:

Andrea Tagliasacchi, Thomas Delame, Michela Spagnuolo, Nina Amenta, Alexandru C Telea. 3D Skeletons: A State-of-the-Art Report. Computer Graphics Forum, 2016, 35 (2), pp.573-597. 10.1111/cgf.12865 . hal-01300281

HAL Id: hal-01300281

<https://hal.science/hal-01300281>

Submitted on 19 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

3D Skeletons: A State-of-the-Art Report

Andrea Tagliasacchi¹ Thomas Delame² Michela Spagnuolo³ Nina Amenta⁴ Alexandru Telea⁵

¹ University of Victoria, ² Univ. Grenoble Alpes & CNRS (LJK), INRIA, ³ CNR, Istituto di Matematica Applicata e Tecnologie Informatiche,

⁴ University of California Davis, ⁵ Institute Johann Bernoulli, Univ. of Groningen

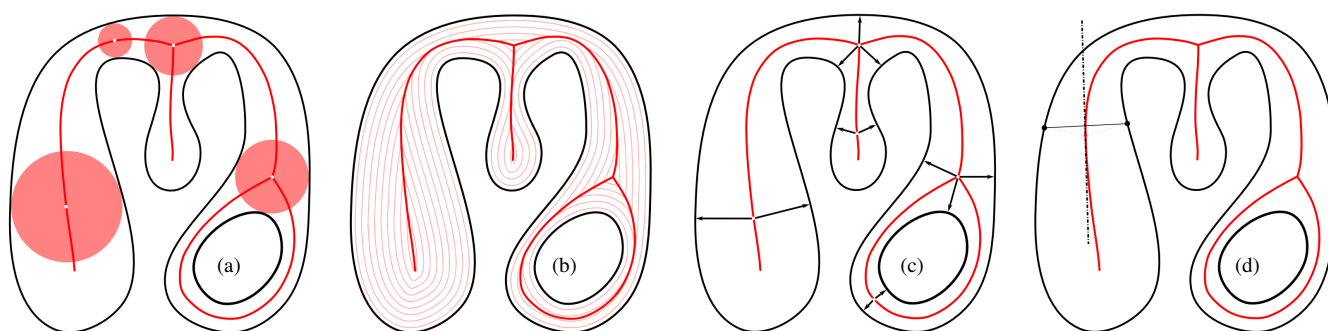


Figure 1: Four alternative definitions of medial skeletons: (a) centers of maximally-inscribed balls; (b) shock graph of the grassfire surface flow; (c) as points with more than one corresponding images on the surface; (d) local axis of reflectional symmetry.

Abstract

Given a shape, a skeleton is a thin centered structure which jointly describes the topology and the geometry of the shape. Skeletons provide an alternative to classical boundary or volumetric representations, which is especially effective for applications where one needs to reason about, and manipulate, the structure of a shape. These skeleton properties make them powerful tools for many types of shape analysis and processing tasks. For a given shape, several skeleton types can be defined, each having its own properties, advantages, and drawbacks. Similarly, a large number of methods exist to compute a given skeleton type, each having its own requirements, advantages, and limitations. While using skeletons for two-dimensional (2D) shapes is a relatively well covered area, developments in the skeletonization of three-dimensional (3D) shapes make these tasks challenging for both researchers and practitioners. This survey presents an overview of 3D shape skeletonization. We start by presenting the definition and properties of various types of 3D skeletons. We propose a taxonomy of 3D skeletons which allows us to further analyze and compare them with respect to their properties. We next overview methods and techniques used to compute all described 3D skeleton types, and discuss their assumptions, advantages, and limitations. Finally, we describe several applications of 3D skeletons, which illustrate their added value for different shape analysis and processing tasks.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid and object representations

1. Introduction

Three-dimensional models of solid shapes are used in many disciplines, including computer graphics, medical imaging, computer aided design, visualization, digital inspection, metrology, and robotics. Such models are usually described using implicit or explicit representations of their surface. *Implicit* (volumetric) representations use a labeling of the (densely) sampled 3D space in which the model is embedded to mark sample points as being inside or outside the shape [BBB*97]. *Explicit*, (surface) representations capture the interface between shape interior and exterior, based on various

sampling and reconstruction schemes, e.g., point clouds or polygon meshes [BKP*10, Ago05]. Both above representations efficiently and effectively support many tasks, such as modeling, processing, and rendering shapes, and are as such ubiquitous in many applications. However, they also have several challenges, as follows.

1.1. Compactness

High-resolution 3D shape representations shapes can be highly expensive in storage and processing terms. While these are needed for tasks like high-fidelity rendering and 3D printing, other tasks, such as shape retrieval, only require access to a specific subset of the

shape properties. Such tasks are favored by a compact shape representation that encodes the key properties for the tasks at hand in a computationally efficient way.

1.2. Expressiveness

Different shape analysis and processing applications focus on different properties of a shape. Examples are shape topology and symmetry (for shape retrieval) or shape thickness (for metrology applications). Modeling and animation applications need to intuitively support complex forms of interactive shape manipulation. All such applications benefit from using shape representations beyond the classical volumetric and surface ones.

Skeletons are one such alternative representation. Informally, skeletons are descriptors which jointly describe the geometry, topology, and symmetry properties of a shape in compact and intuitive ways, providing a mean to capture the ‘essence’ of a shape. The concept originated with medial skeletons in 2D shape understanding, as a way to reduce the large amount of data carried by a shape down to the key information that can be more readily assimilated [Blu67]. An example of a medial skeleton for a 2D shape is shown in Fig. 2a. The concept was next extended to 3D shapes, yielding a wide family of variations, including surface skeletons [SBTZ02], curve skeletons [CSM07], and centerlines [WLK*02, AB02].

Given the long history of skeletonization, understanding the properties of 2D shape skeletons, and computing them efficiently, is a well covered field [SP09, ABE09]. This is quite different for 3D skeletons, for several reasons. First, 3D shapes admit a much richer, and more complex, set of skeleton types, each having specific properties. Secondly, computing 3D skeletons accurately and scalably for large and complex 3D models is much more challenging than computing 2D skeletons. Finally, 3D skeletonization is a newer research area as compared to 2D skeletonization, with many methods and applications having emerged recently. While a few surveys on 3D skeletons exist, they focus on specific problems and/or methods: Cornea *et al.* firstly discuss, to our knowledge, eight desirable properties of curve skeletons, and qualitatively compare four skeletonization methods against these properties [CSM07]. Sobiecki *et al.* [SYJT13] qualitatively compare six contraction-based curve skeletonization methods against six of the properties [CSM07]. Their work is extended in [SJT14] to qualitatively and quantitatively compare four surface and six curve skeletonization methods against six properties. Finally, Saha *et al.* present a survey of voxel-based skeletonization methods and discuss three quality criteria [SBdB15]. The current skeletonization arena is, yet, far broader than the above works cover. Hence, a broad survey of state-of-the-art 3D skeletonization theory, skeleton types, computation methods, and applications is highly due.

1.3. Survey outline

In this paper, we present such a survey of 3D skeletonization. Main differences with respect to existing skeletonization surveys are as follows: We focus solely on 3D skeletons (in contrast to [SBdB15, SP09]); and we cover both surface and curve skeletons using both mesh-based and voxel-based representations (in contrast to [CSM07, SP09, SYJT13, SJT14]). We start by discussing the various definitions for 3D skeletons known in the literature, and the related key properties implied by these definitions. These properties offer key insights to explore the differences between *formal*

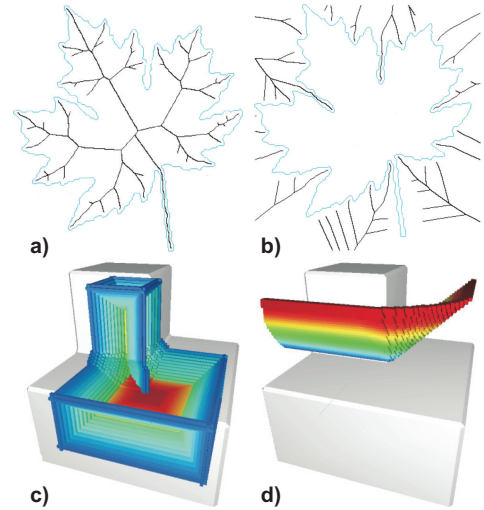


Figure 2: Skeletons (a,c) and their complements (b,d) for 2D shapes (a,b) and 3D shapes (c,d).

skeletons and *practical* skeletons computed by various skeletonization techniques (Sec. 2). Skeletonization techniques, together with their assumptions, advantages, and limitations, are discussed next in Sec. 3. Skeleton analysis and postprocessing operations, which are essential to further construction of skeleton-based shape analysis and processing applications, are covered in Sec. 4. Open challenges to 3D skeletonization are discussed in Sec. 6.

1.4. Notation

An object, or shape is a compact spatial subset $\mathcal{O} \subset \mathbb{R}^n$ with a 2-manifold boundary $\mathcal{S} = \partial\mathcal{O}$, with a focus on three-dimensional shapes ($n = 3$). In \mathbb{R}^2 , we refer to \mathcal{S} as the shape’s *contour*; the medial skeleton of \mathcal{O} is called the medial axis or 2D skeleton. In \mathbb{R}^3 , \mathcal{S} will be referred to as the *surface* of \mathcal{O} ; the corresponding medial skeleton of \mathcal{O} is called the medial surface or surface skeleton. Medial skeletons can be analogously defined for both \mathcal{O} and its complement space $\mathbb{R}^n \setminus \mathcal{O}$, as shown in Figure 2. For discussion clarity, we focus next on discussing the skeleton of \mathcal{O} also known as the internal, or foreground, skeleton. Finally, for the sake of clarity, we will at times depict various skeleton-related properties using 2D examples rather than 3D ones. In all such cases, the discussed properties are identical in 2D and 3D.

2. Theoretical foundations

From their first appearance in shape understanding, skeletons have known many *definitions* (see Sec. 2.1). These definitions are, usually, equivalent, *i.e.*, they imply a unique ‘formal’ skeleton for a given shape \mathcal{O} . This means that skeletons have several well-defined *properties* that can be inferred by studying their definitions, which are essential for the design of shape processing and analysis applications. Yet, the practical application of the above-mentioned definitions leads to fundamentally different *methods* for computing skeletons. In turn, these methods will compute skeletons whose properties approximate the formal-skeleton properties up to various degrees. Additionally, the practical computation of skeletons incurs a number of desirable properties. Existing skeletonization methods satisfy these properties to different degrees. Properties of both formal skeletons and skeletonization methods are discussed in Sec. 2.2.

2.1. Medial Skeletons

The medial skeleton of a shape knows several equivalent definitions, as follows.

2.1.1. Maximally inscribed balls

The original idea of medial skeletons was introduced in 2D by Blum [Blu67]. Here, the skeleton of an object \mathcal{O} was defined as the locus of centers of maximally inscribed balls in \mathcal{O} (see Fig. 3). By adding the radii \mathcal{R} of the corresponding balls to the set of ball centers \mathcal{M} , we obtain the so-called *Medial Axis Transform* $\text{MAT}(\mathcal{O}) = (\mathcal{M}, \mathcal{R})$. A pair $(\mathbf{x}, r) \in \text{MAT}(\mathcal{O})$ consisting of a ball center and corresponding radius is also called a *medial atom*. The corresponding ball $B(\mathbf{x}, r)$ of center \mathbf{x} and radius r is also called a *medial ball*. Atoms are the constitutive elements of a skeleton. Their positions and radii capture a shape's geometry, while their spatial neighborhood relations capture a shape's topology. Atoms, or their positions \mathbf{x} , are also called skeleton points or medial points.

Definition 2.1 The Medial Axis Transform $\text{MAT}(\mathcal{O})$ of \mathcal{O} is the set of centers \mathcal{M} and corresponding radii \mathcal{R} of all maximal inscribed balls in \mathcal{O} .

Since Def. 2.1 only considers *maximal* balls, many balls which are inscribed in \mathcal{O} , but are not maximal, are discarded. Hence, the

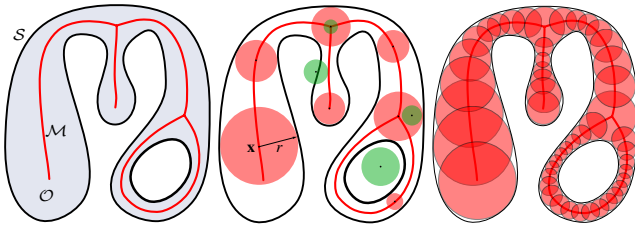


Figure 3: (a) The MAT skeleton \mathcal{M} of the shape \mathcal{O} with contour S . (b) Examples of maximally inscribed balls (red), a medial atom (\mathbf{x}, r) , and balls which are neither maximal nor inscribed, thus not contributing to \mathcal{M} (green). (c) Approximate reconstruction of \mathcal{O} by the union of balls $B(\mathbf{x}, r)$ given by a sparse sampling of \mathcal{M} .

set $\mathcal{M} \subset \mathbb{R}^n$ is sparse – it has constructs with at most $n - 1$ dimensions. These constructs are called medial axis branches (for $n = 2$) or medial axis sheets (for $n = 3$). In 2D, the MAT is a 1D structure consisting of a set of curves (branches) locally centered within the contour S of \mathcal{O} . In 3D, the MAT yields a medial surface, consisting of a set of intersecting manifolds (sheets) with boundary.

As it jointly captures shape geometry and topology, the MAT fully encodes \mathcal{O} . Indeed, \mathcal{O} can be fully reconstructed by computing the union of MAT medial balls, i.e., $\mathcal{O} = \bigcup_{(\mathbf{x}, r) \in \text{MAT}(\mathcal{O})} B(\mathbf{x}, r)$. Hence, the MAT is an invertible transform, i.e., $\mathcal{O} = \text{MAT}^{-1}(\mathcal{M}, \mathcal{R})$. Computing the MAT of a shape \mathcal{O} is called skeletonization. The inverse process of computing \mathcal{O} from its MAT is known as reconstruction or *garbing* (see next Sec. 4.3).

2.1.2. Grassfire analogy

While Blum's original definition [Blu67] provides a simple and solid basis to reason about skeletal properties, thinking intuitively in terms of maximally inscribed balls is difficult. Also, directly applying the MAT definition 2.1 to compute skeletons is not practical. An alternative, and more commonly known, skeleton definition uses

the so-called *grassfire analogy* (Fig. 1b). Imagine $\mathcal{O} \in \mathbb{R}^2$ as a compact patch of grass whose boundary S is set on fire at $t = 0$. The fire propagates isotropically from S towards the interior of \mathcal{O} with uniform speed along the inward normals \mathbf{n} of S . At certain locations, fire fronts coming from different parts of S will meet and quench, thus defining a shock graph [KTZ95].

Definition 2.2 The Medial Axis Transform of \mathcal{O} with boundary S is given by the shock graph of the motion $\dot{S}(t) = -\mathbf{n}(t)$ and the time t when a shock is formed.

This definition not only stands at the core of many skeletonization algorithms (see Sec. 3), but also allows us to intuitively see why such skeletons are called *medial*: As different parts of S move at the same speed, their meeting (quenching) points are at equal distances from S , thus in the local shape center.

A skeleton definition strongly related to the grassfire model can be given in terms of arrival time (Fig. 4a). Consider the motion of S in the direction of its inward normals with unit speed. The position of $S(t)$ at time $t > 0$ can be implicitly described as the level set $\{\mathbf{x} \in \mathcal{O} \mid T(\mathbf{x}) = t\}$ of the arrival-time function $T : \mathbb{R}^n \rightarrow \mathbb{R}^+$ which is the solution of the Eikonal equation

$$\|\nabla T\| = 1 \quad (1)$$

with initial conditions $T(\mathbf{x} \in S) = 0$. The inward normal \mathbf{n} of $S(t)$ at any $\mathbf{x} \in S(t)$ is equal to the gradient $\nabla T(\mathbf{x})$. Hence, the skeleton of S can be defined by the local maxima (ridges) of T or, equivalently, the singularities of ∇T (Fig. 4b). Moreover, the weak solution of Eqn. 1 is the well-known distance transform $DT_S : \mathcal{O} \rightarrow \mathbb{R}^+$ of S

$$DT_S(\mathbf{x} \in \mathcal{O}) = \min_{\mathbf{y} \in S} \|\mathbf{x} - \mathbf{y}\|, \quad (2)$$

where $\|\cdot\|$ denotes the Euclidean metric on \mathbb{R}^n , which can be efficiently computed in linear time [FH12]. Since $\nabla \cdot \nabla T = 0$, skeletons can be defined as the points of non-zero divergence of the distance transform [SBTZ02]. Together, the above definitions of skeletons in terms of local maxima, gradient singularities, or non-null divergence of the distance transform or Eikonal equation solution spawn a multitude of skeletonization methods (Section 3).

2.1.3. Maxwell set

As the grassfire propagates isotropically, quench points are always equidistant from S . Hence, medial points are associated with a least two (Euclidean) closest points on S . This property lies at the core of the *Maxwell set* definition of the MAT [Mat83] (see Fig. 1c).

Definition 2.3 The Medial Axis Transform associates to a shape \mathcal{O} the set of locations $\mathcal{M} \in \mathcal{O}$ with more than one

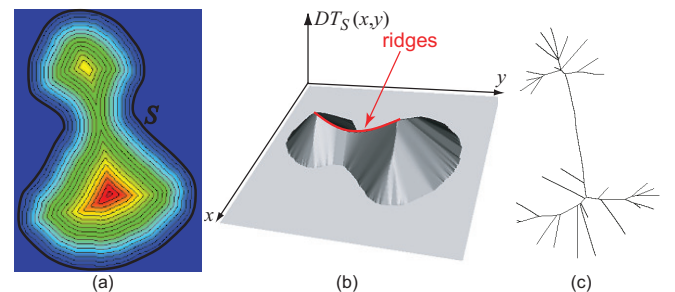


Figure 4: (a) 2D shape boundary S with its distance transform shown by color-coding and level sets. (b) Ridges of the distance transform plot. (c) Corresponding shape skeleton.

corresponding closest point on the boundary \mathcal{S} of \mathcal{O} and their respective distances \mathcal{R} to \mathcal{S} .

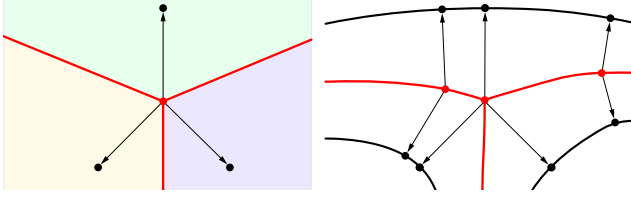


Figure 5: (left) Voronoi diagram (red) encodes locations equidistant from (at least) two input points (black). The medial skeleton (right) extends the Voronoi diagram to freeform curves by encoding the loci being equidistant from at least two points on the curves.

Note that we can rewrite this definition formally using the definition of the shape's distance transform (Eqn. 2) as

$$\text{MAT}(\mathcal{O}) = \{(\mathbf{x} \in \mathcal{O}, DT_{\mathcal{S}}(\mathbf{x})) | \exists \mathbf{y}_1 \in \mathcal{S}, \mathbf{y}_2 \in \mathcal{S}, \mathbf{y}_1 \neq \mathbf{y}_2, \|\mathbf{y}_1 - \mathbf{x}\| = \|\mathbf{y}_2 - \mathbf{x}\| = DT_{\mathcal{S}}(\mathbf{x})\}. \quad (3)$$

Definition 2.3 and its distance-transform based formulation (Eqn. 3) link the definition, and computation, of skeletons with those of Voronoi diagrams (see Fig. 5), as we discuss next in Sec. 3.1.1.1.

2.1.4. Symmetry Set

In an object, symmetry is present at different levels, leading to different ways to abstract shape. For simple shapes (\mathcal{O}_a , Fig. 6a), a single symmetry axis captures the shape's global symmetry. By performing a simple quasi-rigid articulation of \mathcal{O}_a , we obtain \mathcal{O}_b , where a global symmetry axis is insufficient (Fig. 6b).

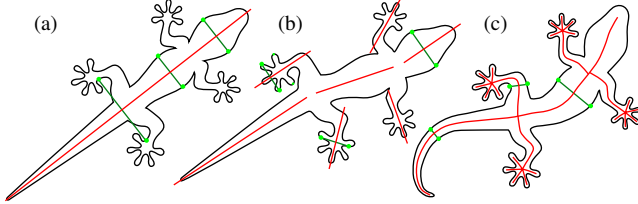


Figure 6: Three planar shapes \mathcal{O}_a , \mathcal{O}_b and \mathcal{O}_c and their (a) global, (b) piecewise, and (c) local symmetry axes. Planar symmetry relations (green lines) between a few point-pairs (green dots) are highlighted. The medial axis encodes local reflectional symmetry (c). Image from [Tag12].

To address this, we could use multiple, more localized, symmetry axes. Yet, when a shape undergoes even complex articulation types, the use of a limited symmetry-axes set becomes too restrictive (\mathcal{O}_c , Fig. 6c). A solution to this is to consider symmetry at its finest level: Each point-pair on the shape boundary is linked by an infinitesimal symmetry relationship; their symmetry centers can be linked together to form a curvilinear symmetry-axis, commonly referred to as the shape skeleton. Let us momentarily consider only shapes with C^1 continuous boundaries – that is, without infinitely sharp corners or edges. For any point $\mathbf{x} \in \mathcal{O}$, the closest boundary point $\mathbf{s}^*(\mathbf{x}) \in \mathcal{S}$ to \mathbf{x} defines a ball $\mathcal{B}_{\mathbf{x},r}$ centered at \mathbf{x} and of radius $r = \|\mathbf{x} - \mathbf{s}^*(\mathbf{x})\|$. This ball can be proved to be tangent to \mathcal{S} at \mathbf{s}^* [ABK98]. If \mathbf{x} is a medial point, then its associated medial ball will be bi-tangent to \mathcal{S} (see Fig. 1d). As such, we can formulate the following

Definition 2.4 The Medial Axis Transform associates to \mathcal{O} the set of centers \mathcal{M} and radii \mathcal{R} of all inscribed balls in \mathcal{O} which are bi-tangent to its boundary \mathcal{S} .

The concept of locus of bi-tangent balls was introduced by [GB85] under the name of *symmetry set*. Hence, medial points can be seen as a subset of the symmetry set, where only *inscribed* balls are considered (see Fig. 6). This observation underlines how skeletons can be thought of as a representation of symmetry, *i.e.*, infinitesimal symmetry axes in \mathbb{R}^2 and infinitesimal symmetry sheets in \mathbb{R}^3 , respectively. Furthermore, we note that while the above definition requires smooth boundaries, it extends to C^0 boundaries by considering bi-tangency in the general sense (*i.e.* using subderivatives).

2.2. Skeleton Properties

The definitions from Sec. 2.1 introduce the Medial Axis Transform, and its corresponding skeletons, in terms of several equivalent definitions. Following, the MAT can be seen as a *dual* representation which captures all shape aspects that a volumetric or boundary representation captures. The key advantage of skeletons, thus, appears when they allow a simpler, more intuitive, and/or computationally effective way to analyze and/or change certain shape properties, than classical volumetric or boundary representations do. To know which such shape aspects skeletons can efficiently and effectively capture, we must first discuss their general properties. We first overview the properties that the formal skeleton definitions (Sec. 2) imply. These are fundamental to any skeletal representation, and thus apply to skeletons produced by any skeletonization method (Sec. 3). However, skeletonization methods have various assumptions, and perform various approximations (Sec. 2.3). Hence, the resulting skeletons will fulfill the theoretical properties up to varying degrees. Separately, to be practical, skeletonization methods themselves should comply with several desirable properties. Properties of practical skeletonization methods are discussed in Sec. 2.4.

2.2.1. Topology encoding

Skeletons have the same homotopy as the shapes they are extracted from *i.e.*, they have the same number of connected components and holes (in 2D), and the same number of connected components, voids (cavities), and tunnels (in 3D) [Lie03]. Also, as we have seen in Sec. 2, skeletons are one dimension smaller than their corresponding shape, in the generic case. For 2D shapes, this yields one-dimensional skeletons, which can be effectively encoded as planar graphs, whose edges represent skeletal branches, and nodes represent junctions of at least three sub-branches. In turn, such graphs are highly convenient tools to support operations such as shape segmentation, comparison, matching, and retrieval. Indeed, skeletal graphs capture the coarse-scale structure, or topology, of shapes, which is key to the above applications, and factor out geometry information, which is less important in those contexts. For 3D tubular shapes having local axial symmetry, medial skeletons exhibit the same one-dimensional structure as 2D skeletons, and thus can be readily used to create skeletal graphs for 3D shape matching applications [SSGD03]. In contrast, the medial skeletons of generic 3D shapes are two-dimensional structures. Such structures can be further simplified to extract a topology-encoding graph [LK07, RvWT08a, JKT13, JST15]. Alternatively, one-dimensional skeletons can be directly extracted from 3D shapes by specialized methods. These so-called curve skeletons, and their properties, are separately discussed in Section 3.2.

2.2.2. Smoothness

For a shape $\mathcal{O} \in \mathbb{R}^n$, \mathcal{M} forms a $(n-1)$ dimensional Whitney stratified set [Mat83]. That is, \mathcal{M} is a finite filtration of closed smooth manifolds $\mathcal{M} = M_{n-1} \supset M_{n-2} \dots \supset M_0$, so that $M_i \setminus M_{i-1}$ is either a smooth i -dimensional sub-manifold of \mathcal{M} or is empty. Each connected component of $M_i \setminus M_{i-1}$ is called an i -dimensional stratum. Strata satisfy the Whitney conditions: Intuitively put, all points on a stratum have neighborhoods that are homeomorphic.

2.2.3. Skeletal structure

As outlined in Section 2, the MAT (\mathcal{M}, R) can be used, with no further changes or analysis, to fully reconstruct \mathcal{O} . While this is useful, more complex applications such as shape segmentation, denoising, deformation, or matching require a more detailed analysis of the *structure* of the MAT. As stated earlier, 2D skeletons have a relatively simple structure. They consist of a set of 1D curve segments, bounded by junction points (where at least three such curves meet) or by endpoints (which correspond to positive local maxima of the curvature of \mathcal{S}). The structure of 2D skeletons is well-understood [PSS*03], and less relevant to our 3D skeleton focus. In three dimensions, medial surfaces consist, in the generic case, of a set of intersecting manifolds with boundaries. Curves where (at least) three such manifolds intersect are also called Y-intersection curves [Dam06, GK04, LK07, CLK09]. Similarly to the 2D case, the boundaries of skeletal manifolds which are not part of Y-intersection curves correspond to curvature maxima, or convex edges, on \mathcal{S} . The generic local structure of \mathcal{M} for medial surfaces received explicit geometric description [Gib00]:

- 2-dimensional strata;
- 1-dimensional strata consisting of *i*) Y-junction curves along which three strata meet in a Y-branching pattern and *ii*) edge curves composed of edge points of \mathcal{M} ;
- 0-dimensional strata consisting of *i*) fin points and *ii*) 6-junction points, where six medial sheets meet along with 4 Y-junction curves.

The above classification was further refined to describe surface skeletons in terms of the number of contact (tangency) points, and tangency order, of maximally inscribed balls with \mathcal{S} [GK04]. Four types of points are described (Fig. 7): Points inside skeletal manifolds (A_2^1 points) have precisely two contact points; points on Y-intersection curves of $k \geq 3$ manifolds (A_k^1 points) have k contact points; points where at least three Y-intersection curves meet (A_4^1 points) have 4 contact points; and points on manifold boundaries (A_3 points) have a finite-size circular-sector or circular-arc of contact. The structure formed by A_4^1 and A_3^1 medial points, *i.e.* the one-dimensional Y-intersection curves and their junctions, forms the so-called *medial scaffold* [LK07].

The structure of \mathcal{M} can also be described differently: The closure of a 2-dimensional stratum is called a *surface component*, which is a possibly non-orientable 2-manifold with boundary. The closure of edge curves that are not boundaries of a surface component are *curve components*. The closure of Y-junction curves are the skeleton junctions, *i.e.*, the locations where two skeletal curve or surface components meet. Medial points whose neighborhood is homeomorphic to a closed half-disk or closed interval are the skeletal borders.

Classifying medial points has many applications, such as robust

detection of edges on 3D shapes [RJT08, KJT15], part-based and patch-based shape segmentation [LK07, RT08a, RT08b], and surface reconstruction from point clouds [CLK09]. Other uses of the medial point classification are discussed in Section 3.2.1.1.

2.2.4. Surface-skeleton correspondence:

As outlined above, skeletons capture the topology of shape, allowing one to reason about the number of connected components, holes, tunnels, and cavities (voids) thereof. At a finer level skeletons also capture the part-whole structure of shape. Simply put, each skeleton branch (in 2D) or manifold sheet (in 3D) corresponds to a different part of a shape; and each junction point (in 2D) or Y-intersection curve (in 3D) corresponds to different parts meeting on the shape. Hence, the skeleton structure naturally induces a shape segmentation.

For the above to work, we need a *mapping* between medial points and points on the surface \mathcal{S} . As outlined by Def. 2.3 and Eqn. 3, medial surface points have at least two different closest-points on \mathcal{S} . Such contact points are described by the so-called feature transform $FT : \mathcal{O} \rightarrow \mathcal{P}(\mathcal{S})$ [MQR03, HR08], defined as

$$FT(\mathbf{x} \in \mathcal{O}) = \{\mathbf{y} \in \mathcal{S} \mid \|\mathbf{x} - \mathbf{y}\| = DT_{\mathcal{S}}(\mathbf{x})\} = \arg \min_{\mathbf{y} \in \mathcal{S}} \|\mathbf{x} - \mathbf{y}\|, \quad (4)$$

which delivers the set of closest points \mathbf{y} on a shape's surface \mathcal{S} to a given interior point \mathbf{x} . These are also called feature points [ST04, HR08] or image points [SP09]. The so-called *spoke vectors* $\mathbf{y} - \mathbf{x}$ [PFJ*03] are parallel with the shape's distance-transform gradient $\nabla DT_{\mathcal{S}}$, which gives, by definition, the shortest paths from \mathcal{S} to the skeleton. As $DT_{\mathcal{S}}$ is divergence-free away from the skeleton (Sec. 2.1.2), spokes do not intersect outside the skeleton.

Using the feature transform (Eqn. 3), Giblin and Kimia's skeletal point classification [GK04] can be elegantly written in terms of the cardinality of the feature transform [RvWT08b, KJT15]. Additionally, by applying Eqn. 4 to skeletal components separated following point classification, we can readily produce various types of shape segmentation, *e.g.* patch-based segmentation from surface-skeleton manifolds [LK07, RT08b] (Fig. 8a,b) or part-based segmentation from curve-skeleton branches [ATC*08, RT08a] (Fig. 8c).

The opposite mapping (from the skeleton to the surface) is also pos-

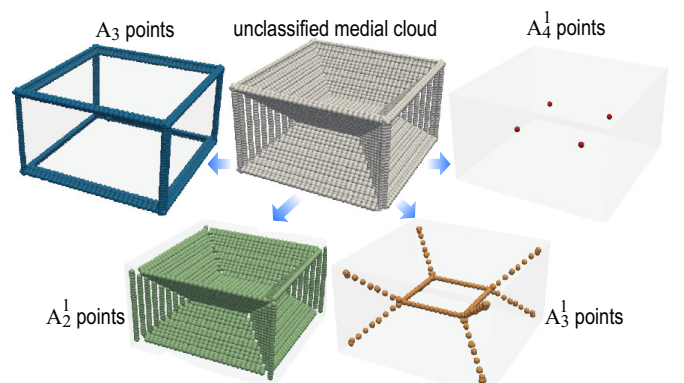


Figure 7: Medial surface point classification based on contact order and count, illustrated for a box shape. Image generated with the method in [KJT15].

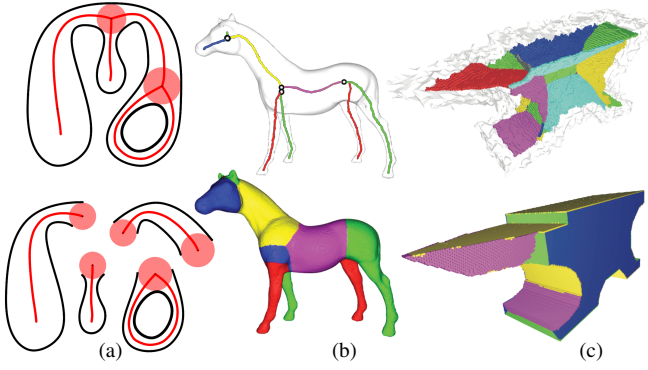


Figure 8: Part-based shape decompositions induced by (a) a 2D skeleton and (b) a 3D curve skeleton. Patch-based decomposition induced by a 3D surface skeleton (c). Colors in (b) and (c) indicate different parts of the skeleton (top row) and shape (bottom row).

sible, and useful, too. In the continuous case, for C^1 smooth surfaces, each point $y \in \mathcal{S}$ will have precisely one medial point via FT^{-1} . This allows a simple way to project the scalar function \mathcal{R} defined on the medial points onto \mathcal{S} . This offers a formal way to define and compute local *shape thickness*, which is crucial in many metrology [Geo10, BCAC04], medical imaging [YP03, SNT*10], and shape retrieval [TV08] applications.

2.2.5. Semi-continuity and instability

To enable its practical use in shape processing, it is desirable for the MAT to be Cauchy or Lipschitz continuous. Intuitively, similar shapes (e.g. in terms of Hausdorff distance [dvOS00]) should yield similar MAT's. A second benefit of such continuity is that we can approximate a shape \mathcal{O} by some similar shape \mathcal{O}' whose MAT is much easier to compute practically, and use this MAT as a 'proxy' for \mathcal{O} 's MAT. This property is well known, and widely used, for many shape processing tasks, such as smoothing, filtering, and edge detection [Tau95, CC00].

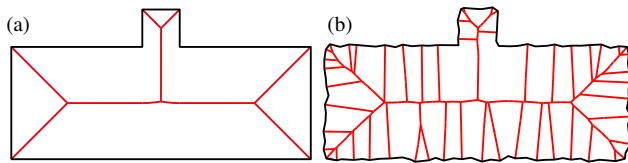


Figure 9: Illustration of semi-continuity. Applying small-scale changes to the shape (a) results in large changes in its skeleton (b).

Unfortunately, MAT's are in general not continuous in the above sense [ABE09]. This means that even very small changes of \mathcal{O} can yield large changes of $\text{MAT}(\mathcal{O})$ (Fig. 9). However, the MAT is *semi-continuous* with respect to the one-sided Hausdorff distance. In detail: Let \mathcal{O}_A be a shape and let $\delta > 0$ be a distance. Semi-continuity means there exists an $\epsilon > 0$ such that for every shape \mathcal{O}_B with $d_H(\mathcal{O}_A^c, \mathcal{O}_B^c) < \epsilon$, we have $d_H(\text{MAT}(\mathcal{O}_A) \mid \text{MAT}(\mathcal{O}_B)) < \delta$, where $\mathcal{O}_A^c = \mathbb{R}^n \setminus \mathcal{O}_A$, $d_H(x \mid y)$ is the one-sided Hausdorff distance of shape x from shape y and $d_H(\cdot, \cdot)$ is the symmetric Hausdorff distance. This means that even if two shapes \mathcal{O}_A and \mathcal{O}_B are very similar, such as the ones in Fig. 9, a subset of the MAT points of one shape can still be far away from any MAT point of the other. Such medial points are typically called *skeletal noise* [FSL04, RvWT08b,

GMPW09] or *spurious points* [SBTZ02, SFM05, PK99b]. As such points are caused by small-scale noise on \mathcal{S} , removing them can make MAT computation Cauchy-continuous, or *stable* with respect to small-scale perturbations of \mathcal{S} . This process is known as *pruning*, or more formally, MAT regularization. A complementary approach to bring continuity to the MAT is to regularize the shape itself, following the semi-continuity property, thus reducing the minimum feature distance, and hence the required ϵ . Both approaches are discussed in Sec. 3.

2.3. Challenges of practical skeletonization

In theory, given an *exact* shape model, and a discrete implementation of any of the skeleton definitions in Sec. 2, we can compute the associated *exact* skeleton. In simple cases, this can be achieved: [FLM03] computed implicit function representations for the sheets of the 3D medial surfaces of polyhedra, [SAR95] combined Voronoi and exact methods. Yet, this is not possible and/or practical to do, for any (complex) 3D shape, for several reasons. To show this, consider the Maxwell-set definition of skeletons (Eqn. 3), one of the most straightforward ways to implement a skeletonization method (similar reasonings hold for the other definitions in Sec. 2). To use this definition in practice, several choices need to be made:

- **Object representation:** Volumetric or boundary representations are clearly the most suited tools for generically representing the shapes \mathcal{O} to skeletonize. However, the specific sampling and interpolation decisions with respect to the surface \mathcal{S} used by such representations affect the definition of $DT_{\mathcal{S}}$. For instance, polygonal models can be skeletonized using bisector methods in 2D [Lee82, Hel01] and in 3D [CKM04]. Many methods allow computing skeletons of voxel shapes [AL01, HR08, ASS11, SBdB15]. Skeletonization can also treat shapes represented using rational curves in 2D [Tzo11] and 3D [MCD11]. Yet, changing the shape representation often requires one to adapt the MAT definition, including the ways to evaluate the distance transform $DT_{\mathcal{S}}$.
- **Boundary points:** We need a way to practically determine when two points on \mathcal{S} are different (term $\mathbf{x}_1 \neq \mathbf{x}_2$ in Eqn. 3). It is well known that spurious skeleton branches are created by closely-spaced point configurations that sample small surface details [OK95, TvW02, CC00, RvWT08b] (Fig. 9). One method to eliminate such undesired skeletal noise is to use a tolerance when comparing surface points [RT07]. A separate reason to use tolerances to compare surface points in Eqn. 3 is that sampled shape surfaces can have widely varying resolutions, so comparison for strict equality is not doable. Yet, using such tolerances can yield too thick and/or noisy skeletons [RT07].
- **Distance estimation:** At the core of the skeleton definition (Eqn. 3) is the distance of an interior point to the surface \mathcal{S} , which can be computed exactly or approximately (terms $\|\mathbf{y}_i - \mathbf{x}\|$ in Eqn. 3). Exact computation is best, but can be expensive for densely-sampled shapes. Distance estimation is also affected by the skeleton representation model: In a volumetric (pixel or voxel grid) model, skeletal-atom positions are *constrained* by this grid. Hence, exact distance estimation, even if possible, yields only a subset of the true skeleton when the fixed grid cannot capture points found at *precisely* equal distances from two different boundary points. Tolerance-based distance estimations and/or comparisons can be used; however, this introduces various deviations from the true skeleton [RT07].

Property	[CSM07]	[SYJT13]	[SJT14]	[SBdB15]
Homotopy	✓	✓	✓	✓
Invariance	✓	✓	✓	✓
Thinness	✓	✓	✓	✓
Centeredness	✓	✓	✓	✓
Smoothness	✓	✓	✓	✓
Details	✓	✓	✓	✓
Regularization	✓	✓	✓	✓
Reconstructibility	✓	✓	✓	✓
Scalability	✓	✓	✓	✓

Table 1: Skeleton(ization) properties covered in survey papers.

Apart from the above concerns, modern 3D skeletonization methods need to be highly scalable, *e.g.*, compute 3D skeletons of complex shapes having millions of voxels or polygons efficiently (*e.g.*, at near-interactive rates) [MBC12, JKT13, JST15]. Such throughputs are highly challenging for most methods.

All above issues converge towards the need to compute *approximate* skeletons, or skeletons of *approximate* shapes. The first option uses the actual input shape but approximates the formal skeleton definitions (Sec. 2) to increase speed, robustness, and ease of skeletonization. The second option modifies the input shape, by using the MAT's semi-continuity, to better fulfill the speed, robustness, and ease of computation requirements. The key common aspect of all practical skeletonization methods is, thus, the use of approximations and/or heuristics when applying the skeletal definitions listed in Sec. 2. These issues are discussed next.

2.4. Properties of Skeletonization Methods

As outlined above, actual skeletons computed by skeletonization methods may differ in various aspects from the formal skeletons implied by the definitions in Sec. 2. For such skeletons to be usable in practice, we need to know how they precisely deviate from their formal counterparts. For skeletonization methods to be useful and usable, they have to comply with several additional non-functional requirements. Both aspects are jointly discussed next.

Defining and assessing desirable properties of skeletons and skeletonization methods has been an important goal in shape processing, see the four key surveys cited in Sec. 1 [CSM07, SYJT13, SJT14, SBdB15]. Table 1 summarizes the key desirable properties covered by these surveys. These are also the properties discussed in most skeletonization papers. Assessing these properties is, obviously, of crucial importance for the selection of a suitable skeletonization method for a specific application context. Since this assessment is quite involved, we discuss it separately next in Sec. 4.1. We next discuss the properties in Tab. 1. For each property, we present its definition and alternative names known in the literature, its relationship with potential approximations introduced by the practical interpretation of skeleton definitions, and its practical implications. For convenience, we employ here the Maxwell-set skeleton definition (Eqn. 3). Yet, all considerations below hold for all other formal medial-skeleton definitions.

2.4.1. Homotopy

Practical skeletons should maintain the homotopy property of their formal counterparts (see Section 2.2.1). While most curve and surface skeletonization methods do that, small-scale tunnels occurring in thin shape parts can be formed and/or discarded by both mesh and voxel-based methods for low-resolution models [SYJT13, SJT14].

In turn, such defects can adversely affect topology-based shape analyses such as shape matching [SSGD03]

2.4.2. Invariance

Since the skeleton definition depends only on the shape \mathcal{O} , and not on its position and/or size in the embedding space, skeletons should be invariant under isometric transforms T of the \mathcal{O} , *i.e.*, $\text{MAT}(T(\mathcal{O})) = T(\text{MAT}(\mathcal{O}))$. Analytic methods (*e.g.* methods which represent both \mathcal{O} and its skeleton as meshes) typically satisfy this property, since all computations are done in high-precision 3D vector space. In contrast, voxel-based methods cannot be fully invariant, since samples of both \mathcal{O} and $\text{MAT}(\mathcal{O})$ are constrained to the fixed voxel grid. The adverse effects of variance can be limited by using exact Euclidean distance transforms [MQR03, HR08] instead of, for instance, approximate *chamfer* distances.

2.4.3. Thinness

By definition, skeletons are infinitesimally thin (Sec. 2.2.3). So, practical skeletons should be as thin as allowed by the space sampling used to model them. Mesh-based skeletons achieve the desired zero thickness by construction. The thickness of voxel-based skeletons is lower bounded by the grid resolution. More importantly, one-voxel thickness conflicts with centeredness (discussed next). Consider the simple example of the 2D axis-aligned rectangle \mathcal{O} whose formal skeleton Fig. 10a shows. In a pixel-based sampling where the rectangle width is even, the skeleton either completely misses the central branch (marked gray in Fig. 10b), or, if constraints are used to ensure homotopy, it will have a two-pixel thick branch covering all gray pixels, or a one-pixel-thick branch which is not perfectly centered within \mathcal{S} . This shows the impossibility to use *exact* distance comparisons for the skeleton definition (Eqn. 3) in a fixed-grid setting.

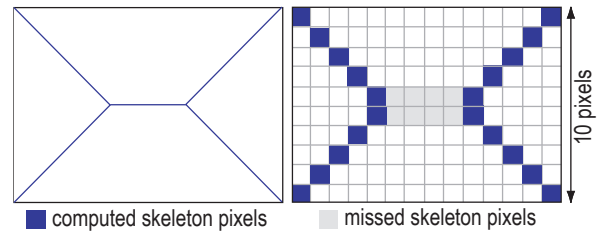


Figure 10: Thickness and centeredness issues. (left) Formal skeleton and (right) its counterpart computed on a fixed pixel grid.

2.4.4. Centeredness

By definition, each skeleton point should be at equal distance from at least two different points of the shape surface \mathcal{S} (Eqn. 3). Several issues exist for practical skeletons with respect to centeredness. First and foremost, centeredness is constrained by the spatial sampling used to represent the skeleton – as discussed above in the context of thickness, voxel-based skeletons cannot always be perfectly centered, even for simple shapes. In contrast, mesh-based surface skeletons can be computed with arbitrarily accurate centeredness [MBC12, JKT13]. A further issue arises for curve skeletons (discussed in more detail in Sec. 3.2): Since there is no *universally-accepted* definition thereof, it is very hard to talk about their formal centeredness. For example, [DS06] and [TAOZ12] (see Eqn. 5) both formally define well-centered curve-skeletons, yet

with very different approaches. In practice, centeredness is crucial when using skeletons for shape reconstruction [ASS11] and metrology [JKT13]. Centeredness can be quantitatively assessed in absolute terms, by measuring distances from skeleton points to their closest surface points; or in relative terms, by comparing two or more skeletons produced by different methods against each other using Hausdorff distance metrics [SJT14, JST15].

2.4.5. Smoothness

Surface skeleton manifolds are known to be at least piecewise C^2 continuous, regardless of the noisiness of the input shape surface \mathcal{S} [PSS*03, SP09]. As such, skeletonization methods should produce piecewise-smooth skeletons. Several problems exist related to smoothness for practical skeletons. First, it is hard to quantitatively assess when a skeleton is sufficiently smooth, since computing a ground-truth skeleton on which smoothness can be assessed is hard or even impossible. Secondly, smoothness can be limited by the sampling resolution and/or sampling model used for the skeleton (*i.e.*, mesh- or voxel-based). The smoothness of mesh-based skeletons strongly depends on the local point density used for sampling. Many such methods have difficulties in densely sampling the so-called *ligature branches* [MSD08], *i.e.*, branches which emerge from the main skeleton towards small-scale convex bumps on the input surface \mathcal{S} [SYJT13, JKT13]. Smoothness can be increased by low-pass filtering the computed skeletal points [ATC*08, HF09, JT12, HWC0*13]. However, unconstrained smoothing can adversely affect centeredness.

2.4.6. Detail resolution

Skeletons should effectively and accurately capture all shape topology and geometry. This property is also known as the ability of skeletons to detect junctions or perform component-wise differentiation of the input shape parts [CSM07, SBdB15], as well as detail preservation [SYJT13]. Detail preservation is important for applications such as exact/global shape matching, retrieval, and reconstruction [CSM07, RvWT08a]. Detail preservation conflicts with semi-continuity and the instability of the MAT (Sec. 2). The key issue here is what is to be considered as a small scale *detail* of a shape *vs* what is to be seen as *noise* – we ideally would like to preserve the former but remove latter. However, as typical to most shape processing applications, the distinction between details and noise is not an easy one to do generically. Most skeletonization methods favor the removal of noise, which is seen as a major problem for using skeletons in practice, even if details are removed, as discussed also next in the context of regularization. For instance, saliency-based methods [TK01, Tel12] propose heuristics to differentiate noise from important details using a local skeleton-*vs*-boundary analysis. However, such methods only work for 2D shapes.

2.4.7. Regularization and LoD

The MAT's sensitivity to small shape changes (Sec. 2.2.5) has major practical implications. Nearly all skeletonization papers mention sensitivity as *the* key challenge in computing usable skeletons. Removing such instabilities is known under various names, *e.g.* pruning [OK95, SP09, ASS11] and regularization [RvWT08a, JKT13]. Regularization can be seen as a filter $F(\cdot, \tau)$ which, when applied to $\text{MAT}(\mathcal{O})$, yields a skeleton $F(\text{MAT}(\mathcal{O}))$ that is Cauchy or Lipschitz continuous with respect to variations of \mathcal{O} smaller than τ . Regularization methods which are also continuous in τ yield progressively simplified skeletons at any user-desired LoD, or a *multi-*

scale skeleton [CC00, FSL04, TvW02, DS06, RvWT08a]. Two types of regularization exist [BGP10, RvWT08a]: local and global.

Local methods include the angle between feature points (Fig. 11d) [ACK01, HR08] and distance-to-boundary [FLM03]. While this approach is able to retain features at different scales, it might result in a skeleton whose topology is drastically different as we change the threshold. The λ -medial axis discards medial samples whose corresponding surface samples have a circumradius larger than λ [CL05a]. For particular choices of λ , the skeleton is homotopic to the shape, and its approximation quality is provably convergent [CL05b]. Unfortunately, this metric does not allow capturing details at different scales, as salient shape features are removed before noise (Fig. 11-c). Other local metrics include divergence measures [BST05, SBTZ02, SFM05] and first-order moments [RT02] of the distance transform. While simple to compute, no local measure can separate locally-identical, yet globally-different, contexts (see *e.g.* Fig. 1, [RvWT08a]). Moreover, many such detectors asymptotically decrease to zero along deep ligature branches, see *e.g.* Fig. 12b which shows the divergence measure in [SBTZ02]). Hence, thresholding such local measures can disconnect skeletons. Reconnection needs extra work and makes pruning less intuitive [SB98].

Global measures monotonically increase from the skeleton boundary $\partial\text{MAT}(\mathcal{O})$ inwards. Thresholding them preserves homotopy. Such measures include the Scale Axis Transform (Fig. 11-e) [BGP10] which exploits union-of-balls (UoB) medial properties [GMPW09]; the medial geodesic function (MGF) measuring the shortest-geodesic path between feature points for curve skeletons [DS06, PH02]; and its extension to surface skeletons [RvWT08a]. These belong to the larger family of so-called ‘collapse metrics’ used to simplify 2D skeletons [CC00, OK95, TvW02]. Collapse metrics are monotonic and have an intuitive geometric meaning: They assign to a skeleton point the amount of shape boundary that corresponds, or ‘collapses to’, that point. Hence, regularization is easy: Thresholding by a value τ removes all skeleton points which encode less than τ boundary-size units. Figure 13 illustrates this by showing progressively simplified 2D and 3D skeletons by three such methods. Yet, for large 3D shapes, computing collapse metrics is not cheap [JST15], even when using GPU parallelization [JKT13].

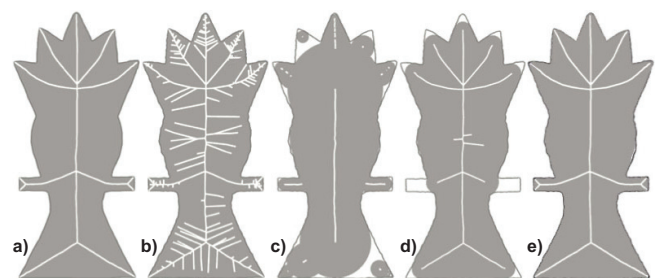


Figure 11: (a) 2D object and its medial axis. (b) Noise added to the boundary yields spurious medial branches. (c) Object angle filtering (λ -medial axis [CL05a]) captures features across different scales but yields topological changes. (d) Distance-to-boundary filtering (γ -medial axis [ACK01]) retains topology but removes small features before spurious branches. (e) The scale axis transform [BGP10] removes noise while retaining small features.

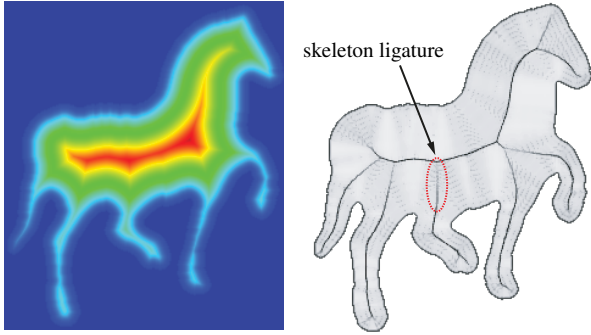


Figure 12: A 2D skeleton can be extracted by (a) computing the distance transform of a shape and then (b) locally detecting high-divergence area [SBTZ02]. (b) Ligature branches get a far lower detector-metric value as compared to other branches.

Regularization can be also done on \mathcal{O} rather than its MAT. Formally, this applies a smoothing filter $F(\cdot, \tau)$ to \mathcal{O} so that $\text{MAT}(F(\mathcal{O}))$ is, generally speaking, Cauchy or Lipschitz continuous upon changes of \mathcal{O} smaller than τ [BGP10]. Yet, controlling the smoothing amount τ to get a desired level of skeletal noise removal is challenging [ABE09].

2.4.8. Reconstructibility

The MAT provides a full encoding of a shape \mathcal{O} and is also fully invertible (Sec. 2). In theory, this allows reconstructing the exact \mathcal{O} from its $\text{MAT}(\mathcal{O})$, a process also known as garbing [DRF12, ASS11, JKT13]. Yet, practical skeletons approximate formal ones, due to approximate representations, sampling limits, or regularization. Hence, exact reconstructibility of a shape from its skeleton is, in practice, rarely possible. High-resolution and accurately-centered mesh-based surface skeletons can efficiently reconstruct shapes up to high detail (Fig. 14). Voxel-based surface skeletons also offer good reconstruction accuracy [ASS11, JST15], albeit with lower quality as compared to mesh methods, due to the discussed limitations of fixed grids.

When collapse metrics (discussed earlier) are used to regularize skeletons by pruning medial points having a collapse value lower than τ , reconstruction has several guaranteed properties: For 2D skeletons, reconstructing the simplified skeleton is equivalent to replacing all corners on the input shape's contour by circular arcs of radius τ [TvW02, SP09] (Fig. 11d). For 3D surface skeletons, this is equivalent to replacing all corners by spherical patches of radius τ , and all edges by cylinder patches of radius τ , respectively [RT02].

2.4.9. Computational scalability

Since their appearance more than forty years ago, skeletons have been used in many applications. 2D skeletons require a relatively limited computing power, as typical input shapes are in the order of 1000^2 pixels. In contrast, modern 3D synthesis and acquisition methods can easily generate shapes up to 1000^3 voxels or, equivalently, millions of surface polygons. As such shapes can be acquired in near-real-time, interactive and scalable 3D skeletonization methods are crucial [CSM07]. To achieve this goal, several approaches have been proposed. Voronoi-based methods typically achieve a complexity of $O(n \log n)$ for n sample points on \mathcal{S} [OK95, DS06, ACK01]. Distance-based methods using voxel sampling achieve a complexity of $O(\mathcal{T} \log \|\mathcal{S}\|)$, where

$\|\mathcal{S}\|$ is the shape boundary length and \mathcal{T} is the average shape thickness [TvW02, FSL04], comparable with the complexity of Voronoi methods. Other methods achieve a linear complexity in the number of input voxels $\|\mathcal{O}\|$ [RT05, HR08], which is slightly higher than [TvW02, FSL04, OK95]. Shape contraction and ball-inscription methods have a complexity of $O(ns)$, where n is the number of surface samples $\|\mathcal{S}\|$ and s is the number of contraction iterations, typically a constant [MBC12, JKT13, ATC*08]; linear-complexity $O(n)$ methods also exist [TAOZ12]. Overall, all such methods are quite similar in theoretical computational costs. However, much can be gained in terms of practical costs, by parallelizing the skeleton-detection operations, e.g. distance transform computation [CTMT10] or ball-inscription method [MBC12, JKT13]. By exploiting the fact that skeletal points, following Def. 2.1, can be searched for independently, GPU-based parallelization can lead to skeleton extraction from mesh models of millions of vertices in near real-time on modern PCs [MBC12, JKT13], however, such high-throughput methods are hindered by complex implementations.

3. Taxonomy of skeletons

Skeletonization methods can be seen as spanning a multidimensional attribute space. A point herein is a method whose attributes describe how the method complies with the properties in Sec. 2.4. An easy-to-use way to present such a space is via a taxonomy tree. Yet, there are many ways to group N methods, each having M attributes, to build such a tree. We address this by assigning tree-levels to attributes that most separate the studied methods. This yields the two-level tree formed by:

Level 1: Dimensionality. The first taxonomy level uses the dimensionality of skeletal parts. For 3D shapes, those can either be 1D curves or 2D surfaces. Hence, methods are subdivided into *curve skeletonization* methods (yielding skeletons having only curve parts) and *surface skeletonization* methods (yielding skeletons generically having surface and curve parts).

Level 2: Sampling. The second level captures the spatial sampling used to represent the input shape and/or its skeleton. Two options exist here: *image methods* use a uniform discrete pixel or voxel sampling of \mathbb{Z}^3 ; *analytic methods* use a continuous, typically non-uniform, sampling into vertices in \mathbb{R}^3 . While analytic methods can be easily converted to image methods by voxelization tools [NT03], the two representations lead to skeletonization methods having quite different properties. Next, when presenting our proposed taxonomy, we refer to the sampled representations of the input shape \mathcal{O} and its surface \mathcal{S} by \mathbb{O} and \mathbb{S} , respectively.

3.1. Surface Skeletons (SS)

These methods aim to produce skeletons that follow the equivalent medial definitions in Sec. 2. Such methods deliver skeletons which have a good correspondence to the input shape, and thereby a high reconstruction power. However, surface skeletons have a complex structure, which makes these methods relatively more complex and slower to compute than curve skeletonization methods.

3.1.1. Analytic Surface Skeletons (ASS)

These methods represent both the input shape \mathcal{O} and its surface skeleton analytically. Hence, they offer the optimal approximation and memory-efficient shape encoding. Yet, they are also arguably the most complex methods in the 3D skeletonization arena.

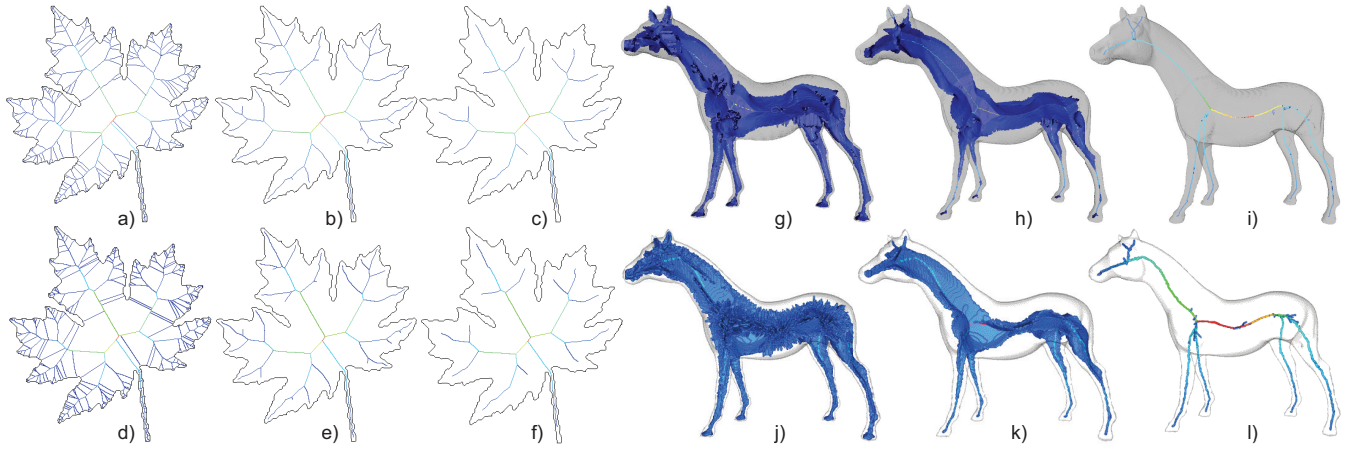


Figure 13: Skeletons progressively simplified by the collapse metrics of [TvW02] (a-c), [JST15] (d-f-j-k), and [RvWT08a] (g-i).

3.1.1.1. Voronoi methods: The Maxwell set form of Def. 2.1 allows computing skeletons in terms of *Voronoi diagrams*. Indeed, the Voronoi diagram $V(\mathbb{S})$ of a sampled surface \mathbb{S} partitions the space into cells $V(x \in \mathbb{S})$ containing the points closer to x than to any other samples. Thus, cell boundaries are points having more than one closest neighbors in \mathbb{S} . By considering the boundaries of Voronoi cells in \mathcal{O} , we get an approximation of the MAT of \mathcal{S} . This is easily seen in 2D, as a Voronoi edge is the locus of points having two closest neighbors on \mathcal{S} . The accuracy of this MAT depends on the sampling of \mathcal{S} (Fig. 15): Given a dense enough δ (uniform) sampling of \mathcal{S} a provably convergent and homotopic medial approximation of MAT(\mathcal{S}) is achievable [BA92, AM97]. This was later generalized to adaptive ϵ -sampling to use less samples in less detailed regions of \mathcal{S} [AB98].

However, the results in [AM97, AB98] do not directly hold for 3D shapes. The issue is that, even for arbitrarily fine samplings, the Delaunay triangulation dual to the Voronoi diagram presents *sliver tetrahedra*. These correspond to Voronoi vertices, which neither fall

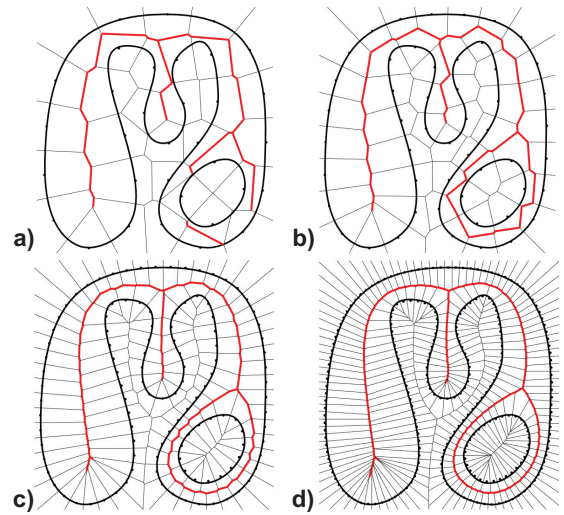


Figure 15: Voronoi diagram of a boundary with increasing and uniform sampling density. Voronoi vertices and edges completely enclosed in the boundary approximate the medial axis. As density increases, the approximation improves. A minimal sampling density is needed to obtain skeletons homotopic to the input shape.

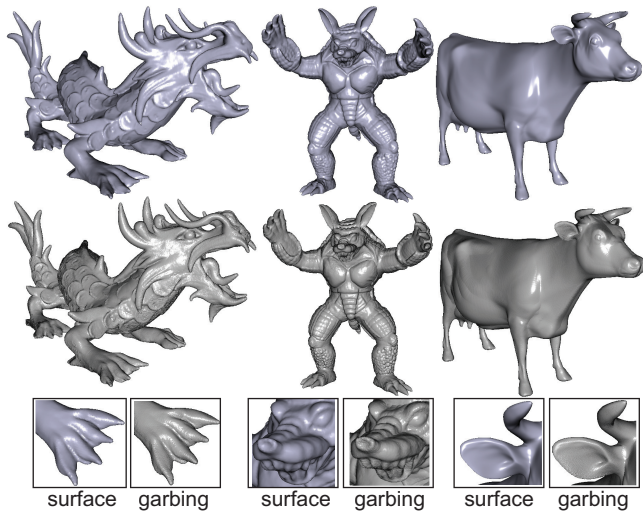


Figure 14: Surface rendering (top row) and skeleton-based reconstruction (middle row). Insets show details [JKT13].

close to the skeleton nor are related to any prominent surface feature (Fig. 16b). To address this, [AB98] approximate the skeleton by only considering the *Voronoi poles* in the diagram, defined as

Definition 3.1 Given a finite set of points $\mathbb{S} \subset \mathbb{R}^n$ which samples a surface \mathcal{S} , and its Voronoi diagram, each $\mathbf{p} \in \mathbb{S}$ is associated with a convex Voronoi polyhedron H . The vertices of H on the two sides of \mathcal{S} which are farthest from \mathbf{p} are the Voronoi poles of \mathbf{p} .

The validity of Voronoi poles for medial approximation was formally verified in [ACK01], by showing that for an ϵ -sampled C^1 manifold, poles approach the surface skeleton as ϵ vanishes. Fig. 16 shows this by comparing the approximation power of poles to the one of Voronoi vertices. Several approaches of linking the poles or other subsets of medial-approximating Voronoi vertices exist, be-

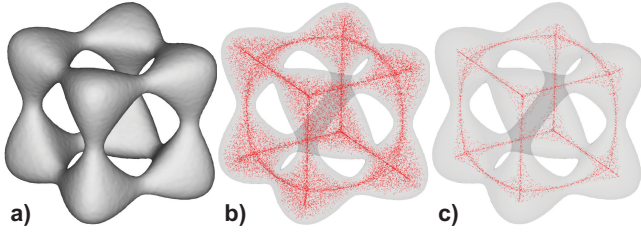


Figure 16: Comparison of medial approximation by interior Voronoi vertices (b) and interior poles (c) of a dense sampling of the shape boundary (a).

sides [ACK01], e.g., filtering the Voronoi diagram [DZ04], and flow-related distance functions [GRS06].

3.1.1.2. Bisector methods are closely related to Voronoi methods. For a point-pair $(\mathbf{p}, \mathbf{q}) \in \mathbb{S}^2$, the bisector $B(\mathbf{p}, \mathbf{q}) \subset \mathbb{R}^3$ is the point-set equidistant from \mathbf{p} and \mathbf{q} . Bisector skeletonization methods [Lee82, CKM04] compute $B(\mathbf{p}, \mathbf{q})$ for all sample-pairs in \mathbb{S} . Bisectors $B(\mathbf{p}, \mathbf{q})$ are pruned next to yield their point-subsets $H(\mathbf{p}, \mathbf{q})$ that are closest to \mathbf{p} and \mathbf{q} . If $H(\mathbf{p}, \mathbf{q})$ is not empty, it is the Voronoi face dual to the Delaunay edge \mathbf{pq} of \mathbb{S} . Restricting $\cup H(\mathbf{p}, \mathbf{q})$ to the inside of \mathcal{O} yields the shape's surface skeleton. While very similar to Voronoi methods, bisector computation is typically far less efficient than Voronoi diagram construction – first, all bisectors are computed with brute force, while most get discarded next. Medial scaffold (MS) methods [LK01, LK03, LK07] address this by using the skeletal-point classification (Fig. 7) to compute only the bisector parts relevant to the surface skeleton. The resulting skeletons are as accurate as those produced by Voronoi methods; while MS methods are, still, slower than Voronoi methods, they also deliver a medial point classification atop the computed skeleton (Sec. 2.2.3).

3.1.1.3. Shrinking ball methods literally apply Def. 2.1 on \mathbb{S} to compute its surface skeleton: For each $\mathbf{x} \in \mathbb{S}$, a ball tangent to \mathbb{S} at \mathbf{x} is iteratively shrunk until it touches another point $\mathbf{y} \neq \mathbf{x}$ of \mathbb{S} . The ball center is then a medial point. Such methods are simple to implement, very accurate, computationally efficient by the use of fast nearest-neighbor schemes to test if a ball is inscribed [AMN*98], and parallelize easily, being the fastest existing surface-skeletonization techniques [MBC12, JKT13]. However, they require a very fine sampling \mathbb{S} of \mathcal{S} to yield (equally) fine skeleton sampling. Also, they deliver only a skeleton point-cloud rather than a meshed model.

3.1.2. Image Surface Skeletons (ISS)

These methods use an image model of both the input shape \mathcal{O} and the produced surface skeleton. For 2D skeletonization, most methods are image-based [SP09] with only a few analytic ones [OK95]. While in general simpler to implement than ASS methods, ISS methods have far larger memory demands – typically $O(n^3)$ instead of $O(n^2)$ for an n -point sampling of the surface $\partial\mathbb{S}$ [SFM05]. Also, they can only capture centeredness within the bounds of the sampling resolution of \mathbb{O} (Fig. 10 for a 2D illustration). Image surface skeletonization is also discussed in recent surveys [SJT14, SBdB15].

3.1.2.1. Topological thinning methods: One of the earliest approaches to compute surface skeletons was iteratively removing

voxels from the boundary $\partial\mathbb{O}$ until a ‘thin’ model is reached. This simulates the grassfire evolution (Def. 2.2): Two fire fronts quench when they reach a *simple point*, i.e., a voxel that cannot be removed without changing the topology of \mathbb{O} [TF81, Pud98, PK99b]. Thinning methods differ mainly in the (a) voxel removal order; and (b) simple point definition. Removal can be done either by alternating thinning between the opposite faces of an axis-aligned bounding box of \mathbb{O} (also called *directional thinning*), which is cheap, but approximates centeredness poorly [PK99b]; or in the order given by the distance transform $DT_{\mathbb{S}}$, which yields better centered skeletons [Pud98, ASS11]. Simple points $\mathbf{v} \in \mathbb{O}$ are usually defined by using their 26-neighborhood $\mathcal{N}_{26}(\mathbf{v})$. For instance, [BM94, SC94] use the 26-connectedness of \mathbb{O} and 6-connectedness for $\overline{\mathbb{O}}$ to define simple voxels \mathbf{v} as those where both $\mathcal{N}_{26}(\mathbf{v}) \cap \mathbb{V}$ and $\mathcal{N}_{26}(\mathbf{v}) \cap \overline{\mathbb{V}}$ have one connected component each. Another way to define simple points uses the collapse operation in a cubical complexes framework [CB08]. Tens of simple point definitions have been proposed, each with its strengths and limitations [SBdB15]. While all such methods preserve topology well, they fare less well in delivering smooth, noise-free, centered skeletons.

Thinning is an iterative process, so speeding it up by parallelizing voxel removal has been a major focus [SBdB15]. If one can independently detect and remove non-simple points, thinning parallelizes trivially to the entire voxel-set \mathbb{O} . While many parallel thinning methods exist, with optimal linear complexity $O(\|\mathbb{O}\|)$, their concrete run-times are not often given in the literature. Also, recent distance-field methods achieve identical complexity with better centeredness and smoothness as discussed below.

3.1.2.2. Distance field methods: These methods exploit skeleton definitions based on the singularities of the distance transform and related fields (Sec. 2.1.2) [KSKB95, GF96, RT02, SBTZ02, HR08]. Computation of exact Euclidean distance transforms and skeletonization can be efficiently implemented on GPUs, making such methods amongst the fastest surface skeletonization techniques [SFM05, vDvdWT06, CTMT10]. Regularization can be next added using local [FLM03] or global [RvWT08b, JST15] metrics. Filtering the singularity detection and/or the input distance field yields smooth skeletons [SBTZ02, RT02]. However, such methods are inherently *local*, which may lead to disconnected, non-homotopy-preserving skeletons. To correct this, singularities are removed only when they are non-simple points [SBTZ02, JST15].

3.2. Curve Skeletons (CS)

Noting that applying the skeleton definitions in Sec. 2 yields very complex structures as compared to 2D-shape skeletons, researchers focused on creating skeletons of 3D shapes which (a) *loosely* follow the desirable properties implied by the formal definitions, and (b) keep the 1D simplicity of 2D-shape skeletons. These so-called *curve skeletons* are loosely defined as 1D structures “locally centered” in a shape [CSM07]. The lack of a unanimously accepted formal definition has led to many methods which compute curve skeletons following different, and typically implicit, definitions. This makes it hard to analytically compare, and reason about, the exact properties of the produced curve skeletons [SYJT13]. However, for tubular shapes having local axial circular cross-sections, curve skeletons can be defined as being 1D structures that, at least, preserve shape topology. This has led to (a) the use of curve skeletons chiefly for such shapes; and (b) the derivation of computa-

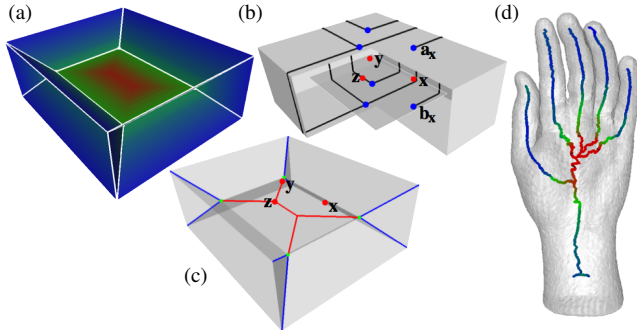


Figure 17: (a) ASS color-coded by the medial geodesic function (MGF). (b) Three points (red) on a cuboid's medial surface; their corresponding image points (blue); shortest paths on the surface between these points (black). (c) high-divergence points define the ACS. (d) ACS of a noisy hand model. Courtesy of [DS06].

tion algorithms based on the assumption that shapes roughly satisfy the above-mentioned local property, and that the delivered curve skeletons should satisfy the observed desirable properties of surface skeletons (Sec. 2.4). Such curve skeletonization algorithms are discussed next. While curve and surface skeletons have at least some common ground in terms of definitions, requirements and properties, other descriptors exist that only share the name of skeleton. Such descriptors are too diverse to be jointly covered in detail in this survey.

3.2.1. Analytic Curve Skeletons (ACS)

These methods represent both the shape \mathcal{O} and its curve skeleton analytically. Hence, they share the accuracy and smoothness typical to ASS, but also their higher implementation complexity.

3.2.1.1. Medial-surface-based methods: Many heuristics have been proposed to compute curve skeletons by pruning surface skeletons to 1D curve. In [DS06], the ACS is computed as the shock graph of the Medial Geodesic Function (MGF), a scalar function defined on the ASS. Given a medial atom $a(\mathbf{x}, r)$ and its two feature points \mathbf{f}_1 and \mathbf{f}_2 , the MGF is defined as $\text{MGF}(\mathbf{x}) = d_S(\mathbf{f}_1, \mathbf{f}_2)$, where $d_S(\cdot, \cdot)$ is the geodesic distance on \mathcal{S} (Fig. 17). The key value of the MGF is that it appears to be monotonically increasing from the ASS boundary to its center (although this property was not formally proven), so the extracted curve skeletons are not only centered within the shape, but also centered within the ASS itself. A variant of the MGF defines curve skeletons as the locus of ASS points admitting two different equal-length geodesics between their feature points [RvWT08a, JKT13]. MGF-based methods provide one of the few formal definitions of curve skeletons. However, this definition has not (yet) been recognized as being universally valuable for all contexts where curve skeletons are needed. Also, such methods are highly expensive, as they require accurate computation of geodesics between all surface point-pairs that are features of ASS points.

3.2.1.2. Generalized field analysis methods: To encode the notion of well-centeredness within the shape, one can replace the Euclidean closest-point metric (Eqn. 3) with a less-localized version [HF09]. The value of this new distance field is computed as the average of potential fields of many boundary samples, and the skeleton extraction is performed by tracing curves seeded at critical points along high-divergence directions. Given the averaging,

such curve skeletons are very smooth. However, they require the expensive computation of the field throughout the full volume \mathbb{O} , as well as complex algorithms to robustly trace the field discontinuities [CSM07].

3.2.1.3. Contraction methods: The 1D curve representation of ACS is a geometric entity with vanishing surface area and therefore vanishing volume. To compute the ACS, one can design a shape evolution, or shape flow, that smoothly evolves the shape boundary into the one of its ACS (see Fig. 18). This ‘contraction’ idea was pioneered in [WL08], where a *volumetric* approach is proposed: The shape’s interior is first discretized into voxels, and a flow is then defined by an energy measuring the volumetric grid’s edge lengths. To avoid this energy from isotropically scaling the contracting shape down to a point, a constrained problem is solved where the locations of voxels close to the shrinking surface are required not to move too far from their initial location on \mathcal{S} . A variant was later proposed in [ATC*08], where the optimization is directly formulated on the shape surface \mathcal{S} . Like in [WL08], constraints are added to avoid the trivial solution and also retain important surface features. Both above constraints, called attraction and contraction, can efficiently be encoded by a system of linear equations for which efficient solvers are available. The fundamental connection between contraction and differential geometry was first revealed by [CK11] and then formalized in [TAOZ12]. This simplified the complex optimization in [ATC*08] to a discretization of the *mean curvature flow*, a well-known flow in differential geometry. The differential equation of mean curvature flow is the Euler-Lagrange of a functional integrating the shape’s surface area. Hence, motion by mean curvature results in the desired surface-area loss through time [RW03]. To adapt such flow to skeletons, the flow is modified so that it stops once the surface is locally fully contracted, *i.e.*

$$\begin{cases} \dot{\mathcal{S}}(t) = -H(t)\mathbf{n}(t) & \text{if } H(t) < \infty \\ \dot{\mathcal{S}}(t) = 0 & \text{otherwise} \end{cases} \quad (5)$$

where $\dot{\mathcal{S}} = \partial\mathcal{S}/\partial t$, H is the mean curvature on \mathcal{S} , and \mathbf{n} is the normal of \mathcal{S} . As curvature motion is a smoothing flow [RW03], skeletons extracted with this procedure are also naturally smooth and resilient to surface noise. Mean curvature is also directly connected to the discrete Laplace-Beltrami operator, leading to the aforementioned computational efficiency. However, this flow is known to degrade the quality of the underlying triangulation \mathbb{S} [TAOZ12]. To fix this, advanced methods that discretize Eqn. 5 by projecting its solution in the embedding space can be used [CK11], or alternatively the quality of the triangulation must be adapted during the motion [TAOZ12]. While these ACS are not naturally centered, the variational formulation of Eqn 5 can be naturally combined with the grassfire interpretation of MAT (Def. 2.2), resulting in well-centered curve skeletons [TAOZ12]. Such contraction methods can also be used to skeletonize shapes represented as unstructured point clouds, if an appropriate discretization of the Laplace-Beltrami operator is used. A first method in this class builds the Laplacian by first projecting the surface point samples on their local PCA tangent plane and then computing their local triangulation [CTO*10]. Other approaches contract point clouds by generalizing robust fairing techniques like *locally optimal projections* (LOP) [LCOLTE07] to robust $L1$ point cloud contraction [HWC0*13]. While such methods provide good performance on raw (outlier ridden) point clouds, whether they admit an elegant variational formulation like the one in Eqn. 5 remains an open question.

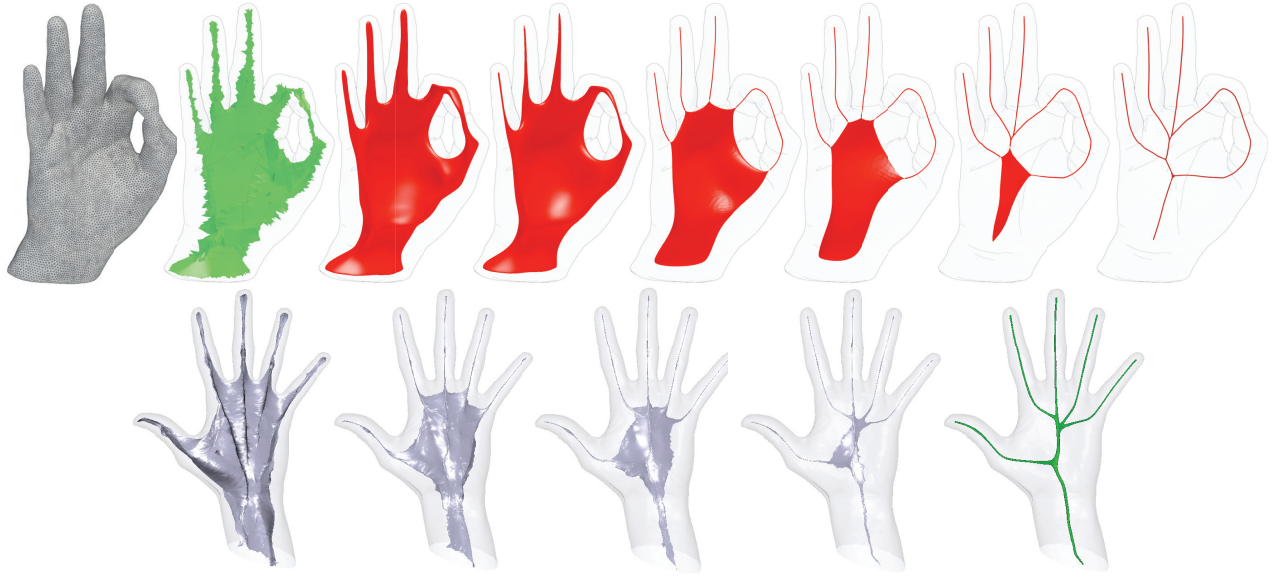


Figure 18: Top row: Surface mesh of a hand model and its surface axis, followed by four contraction steps to compute curve skeletons [TAOZ12]. Bottom row: Five contraction steps of a surface skeleton towards a curve skeleton [JT12].

Other contraction methods start from the surface skeleton (ASS) rather than the input surface S . The method in [JT12] iteratively contracts the ASS by moving its points \mathbf{x} in the direction of the sum of the tangent vectors of the feature-points of \mathbf{x} to S . Since this sum is known to be tangent to the ACS at \mathbf{x} [LK07], contraction effectively takes place within the ASS, thereby maintaining centeredness of the resulting curve skeletons. However, to yield good results, this method requires a finely-sampled ASS; see bottom row of Fig. 18.

3.2.1.4. Mesh decimation methods: Decimation can be used to compute curve skeletons as, in the limit, it yields edges having no incident faces, thus modeling a discrete 1D curve network. This idea is pioneered in [LWTH01] by using an iterative edge collapse on a triangle mesh. While centeredness and smoothness of the produced ACS is suboptimal as compared to other ACS methods discussed in this section, this approach is widely used in post-processing to convert contracted geometry into curve networks [WL08, ATC*08, CTO*10, TAOZ12]. Recent techniques have also tried to improve the embedding quality by regularizing the connectivity before each topological contraction step [JXC*12].

3.2.1.5. Property-grouping methods: As stated earlier, skeletons provide ways to decompose a shape in its fundamental parts and their interconnections (Sec. 2.2). Property-grouping methods invert this process: First, portions of the input shape are grouped into parts; next, a skeleton is generated that describes their spatial configuration. While *reflectional symmetry* (Def. 2.4) can be exploited for skeletonization in 2D [Zhu99], in 3D it is *rotational symmetry* that must be considered [TZCO09] (see Fig. 19b). Conversely, convex decomposition segment a shape into parts whose principal axes (in a PCA sense) are well-centered [LKA06] (Fig. 19a), while in animated articulated models concavities close to bending joints create hints to hierarchically segment the shape [KT03]. Coherence of information over animation time also provides important hints to extract curve skeletons associated with articulated motion. A set of frames where the shape is posed differently can be analyzed to discover the underlying articulated structure. Following this intu-

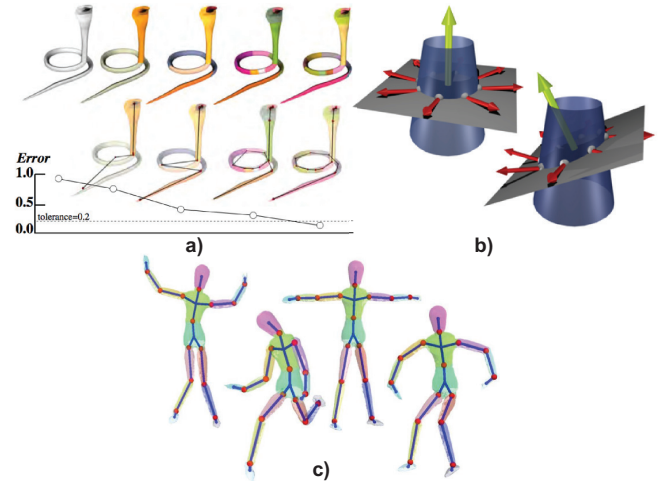


Figure 19: (a) Convex decomposition allows to recycle PCA axes as skeletal branches [LKA06]; the finer the decomposition, the more accurate the skeletonization. (b) Local rotational symmetry of a group of samples can be measured by observing the surface normals arrangement on the Gaussian image [TZCO09]. (c) The articulation of a shape is essentially described by a set of locally rigid transformations [AKP*04]; these transformations can be identified and exploited to create a kinematic skeleton.

ition, [AKP*04, SY07, DATTS08] construct curve skeletons by first clustering parts of the shape undergoing similar transformations, associating a so-called ‘bone’ (1D ACS branch) to each of them and connecting them together to create the ACS (Fig. 19c).

3.2.1.6. Topology-driven methods: A different, very efficient, way to compute curve skeletons uses a *topological* approach: Given a shape \mathcal{O} , consider a real-valued function f defined on the manifold contour \mathcal{S} of \mathcal{O} . The simplicial complex [Ree46], usually

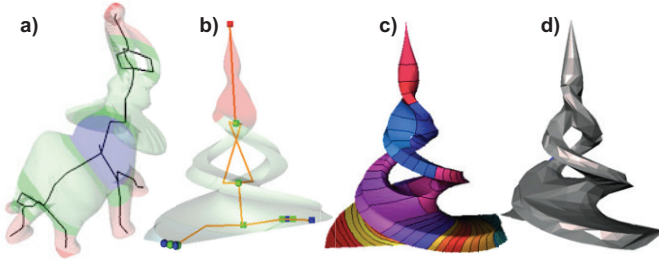


Figure 20: (a) Reeb graph of a 3D shape built by the integral geodesic distance [HSKK01]. Shape reconstruction (d) from its Reeb graph (b) using level set tiling (c) [BPS*10].

known as *Reeb graph*, is the quotient space defined by the equivalence relation that finds points belonging to the same connected component of f 's level sets. Under some hypotheses on S and f , this space is a finite and connected simplicial complex K of dimension 1, so that the counter-image of each vertex of K is a singular connected component of f 's level sets, and the counter-image of each simplex's interior is homeomorphic to the topological product of one connected component of the level sets by \mathbb{R} [Ree46]. Hence, Reeb graph vertices are given by the critical points of f , and its edges represent compact sets of homeomorphic level-set components between critical-point pairs. Reeb graphs can be embedded geometrically to define curve skeletons in a very compact way, not necessarily linked to a reasoning on symmetry (Fig. 20a).

The definition of Reeb graphs nicely formalizes the concise topological encoding of shapes. Various geometric embeddings can be used for the Reeb simplicial complex. The traditional barycentric one maps simplex vertices are mapped to images of f 's critical points on S , and edges to the barycentres of the non-singular level sets of f . This mapping guarantees that the Reeb graph embedding remains internal to the shape and thus gives a well-defined surface-skeleton correspondence. As f is a parameter, different choices for f yield different curve skeleton, which reflect the behaviour of f on S . This gives flexibility in the delineation of a skeletal line, which may be guided to follow specific characteristics, e.g., Reeb graphs defined by the integral geodesic distance [HSKK01] or by Laplace-Beltrami eigenfunctions [BMMP03]. On the other hand, some choices of f may yield skeletons which poorly characterize the shape geometrically. For any f , however, the Reeb graph preserves the shape topology. Concerning other skeleton properties (Sec. 3), the Reeb graph's invariance to isometric transformations of the shape depends on the invariance of f ; thinness is ensured by construction; centeredness depends on the geometric embedding; and detail resolution depends mostly on the smoothness of f . Concerning reconstructability, Biasotti *et al.* [BPS*10] proposed an automatic coding/decoding of 3D shapes using Reeb graphs thanks to an iterative refinement of level sets and their tiling (Fig. 20b-d). Reeb graphs can be extracted following a multi-resolution strategy, as in one of the first papers exploring them in graphics [SKK91]. Recent trends studying the persistence of f 's critical points propose a LOD-like organization of Reeb graphs. Their computational scalability and discretization issues are discussed in [BDF*08].

3.2.2. Image Curve Skeletons (ICS)

Similar to ISS methods, ICS methods use an image model for both the input shape and the produced curve skeleton. They can be or-

ganized along the main categories of ICS methods – thinning and distance-based. However, defining a suitable regularization metric for ICS is harder than for ISS. As such, ICS methods have in general a higher difficulty in being noise-resistant and simultaneously capturing fine-scale shape details. Separately, delivering centered and smooth curve skeletons is hard due to the inherent fixed-grid limitations and the difficulty of suitably defining centeredness for a curve skeleton, both mentioned earlier.

3.2.2.1. Topological thinning methods: Most ICS methods fall in this class. Tools from mathematical morphology [Ser82] were among the first used to compute curve skeletons by thinning. The residue of openings, using Lantuéjoul's formula [Lan79], usually leads to disconnected skeletons; methods based on homotopic thinning [Beu94, PK99a, MBPL99], yield connected skeletons. Yet, pure (directional) thinning cannot ensure centeredness and smoothness [MS96, PK99a]. To favor noise-resistance, Ju *et al.* extract curve skeletons by alternating thinning and skeleton pruning [JBC07]; Liu *et al.* propose a robust thinning method that works on cell-complexes built using voxelization. They introduce a *medial persistence* regularization metric that discriminates shape parts with different anisotropic elongations, e.g., tubes or plates, similar to [JBC07], which can deliver a continuum between surface skeletons and curve skeletons, similar to *meso-skeletons* [TAOZ12].

3.2.2.2. Distance based methods: To increase smoothness and centeredness, thinning can be done in distance-to-boundary order, much like for ISS methods [Pud98, ASS11]. Alternatively, curve skeletons can be extracted directly from the singularities of distance-related fields. Such methods are invariant under isometric transformations. For example, Zhou *et al.* extract the local maxima of the distance transform and connect these by 1D paths to create curve skeletons that are connected, centered, and one-voxel-thin. The surface skeletons delivered by the divergence-based method in [SBT02] can be homotopically thinned to deliver curve skeletons, albeit with detail loss [SJT14]. The MGF metric proposed for ACS [DS06] was extended to compute image-based curve skeletons [RvWT08a]. Extracting curve skeletons as *subsets* of surface skeletons has the important advantage of enforcing centeredness of the former based on the guaranteed centeredness of the latter.

3.2.2.3. Projection methods: A radically different approach to compute 3D curve skeletons is to exploit the high quality and computational speed of existing 2D skeletonization methods. Formally put, let $P: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ be a 'projection' mapping a 3D shape \mathcal{O} to a 2D shape, and let CS^n be the curve skeletonization operation in \mathbb{R}^n . Projection methods compute the 3D curve skeleton $CS^3(\mathcal{O})$ by using suitably computed 2D skeletons $CS^2(P(\mathcal{O}))$. For example, one can use for P the axis-aligned slicing of \mathcal{O} and compute $CS^3(\mathcal{O})$ as the intersection of all $CS^2(P(\mathcal{O}))$ for all slices taken for the three axes [TV03]. Alternatively, one can use for P the 3D-to-2D orthographic projection and compute $CS^3(\mathcal{O})$ as the intersection of all $CS^2(P(\mathcal{O}))$ for a large number of viewpoints uniformly spread around \mathcal{O} [LGS12].

4. Analyzing Skeletons

As outlined in Sec. 1, skeletons are not a means by themselves, but useful *tools* that enable various shape analysis and processing operations. However, skeletons need various analysis and processing steps to make them directly suitable for such usage. The ones that are most frequently used are outlined next.

4.1. Comparing Skeletons

As shown in Section 3, skeletonization methods, and their results, widely differ in many aspects, including their compliance with the desired properties listed in Sec. 2.2. So, before using such methods in practice, a first and important task is to compare them to assess which method is optimally suited to the constraints of a specific application. Assessment is most often done by qualitative means, *e.g.*, visually comparing skeletons obtained with different methods. Figure 21 shows a typical example of such a qualitative comparison of 10 surface and 6 curve skeletonization methods, done by visual inspection. In this example, it is easy to observe that the compared methods satisfy several properties quite differently (homotopy preservation, smoothness, centeredness, robustness to noise, and thinness). This implicitly explains the continuing research efforts in designing better skeletonization methods.

In contrast to qualitative (visual) assessment, quantitative assessment is far less common in skeletonization papers. An exception here is scalability, which is benchmarked by time and/or memory consumption figures. Recently, quantitative assessment was also used for other skeleton properties, *e.g.* reconstructibility [ASS11] and centeredness [SJT14], based on Hausdorff measures between the compared shapes.

Table 2 shows an overview of comparison efforts for a number of recent skeletonization methods. As visible, the extent of comparison efforts is, in general, quite limited. This can be explained by several factors: (a) the difficulty of obtaining (full) implementations of competing methods; (b) the lack of unanimously accepted formal definitions for certain skeletal properties such as robustness to noise, smoothness, regularization, and centeredness (of curve skeletons); (c) the lack of a ‘ground truth’ skeleton (for curve skeletons), stemming from the lack of an unanimously accepted curve-skeleton formal definition; (d) the lack of an established benchmark of 3D shapes exhibiting (all) relevant challenges for surface and curve skeletonization; and (e) the sheer amount of skeletonization algorithms known in the literature. Note that most of the issues above, except (c), also hold for 2D skeletons, which explains why comparisons of 2D skeletons are also not very extensive in the literature. Given the above, benchmarking 3D skeletonization methods against desirable properties and/or against other competing methods is, and will likely remain, hard. This is an increasingly important challenge for both researchers and practitioners involved with 3D skeletal descriptors.

4.2. Reconstructing skeletal structure

To be useful and usable for most practical applications, the ‘raw’ MAT consisting of the medial point cloud and inscribed ball radii needs further analysis. This includes (a) creating compact representations of medial skeletons; and (b) separating skeletons into their composing parts (manifolds for surface skeletons; curve segments for curve-skeletons). Note that (b) is tightly related to finding the boundaries of these parts (external boundaries and Y-intersection curves for surface skeletons; tips and junction points and for curve skeletons), as discussed in Sec. 2.2. Doing both (a) and (b) for image skeletons is relatively simple – these are by definition connected sets of voxels, which addresses (a); and finding intersection points and tips can be easily and efficiently done by using the many templates available in the image morphology literature,

e.g. [BM94, SC94, PK99a], or by using the cardinality of the voxel-based feature transform (Eqn. 4) [RT08c], which addresses (b) (see Fig. 22a). In contrast, doing both (a) and (b) for analytic curve and/or surface skeletons is considerably more complex, as follows.

(a) *Compact skeletons*: Analytic representations of medial skeletons minimally need a raw MAT, defined above. While such point clouds directly support several tasks like garbing (Sec. 4.3 next) or local thickness estimation [JKT13], using them for other tasks like shape segmentation and shape classification needs complicated heuristics [ATC*08, CLK09, KJT15]. Consider for instance the most basic task of *visualizing* a compact and smooth representation of a surface skeleton – this should be easily possible given a raw MAT. This goal is closely related to surface reconstruction from unorganized point clouds, a well-studied problem in computer graphics [BTS*14]. Yet, few reconstruction methods can treat multiply intersecting manifolds with boundaries embedded in noisy point clouds, as surface skeletons are [CLK09, KJT14a], and even such methods have limited robustness. A simple way to create mesh representations of surface skeletons from medial clouds is to *project* the vertices of the input mesh on the skeleton using the inverse of the feature transform (Eqn. 4) [JT12]. This yields meshed surface skeletons of high *visual* quality; yet, these may contain incorrectly oriented and/or self-intersecting triangles and duplicate vertices. Hence, this method is mainly useful for visualization purposes.

(b) *Segmenting skeletons*: At the core of segmenting analytic surface-skeleton models is the classification of medial points [GK04, LK07] which can be done via the medial scaffold [LK07] (Fig. 22b) or by using the cardinality of the feature transform (Eqn. 4) [KJT15]. The medial scaffold approach is slower and more complex to implement, but delivers a surface skeleton *segmentation* into manifolds. In contrast, the feature transform approach is simpler to implement, and faster, but delivers only a medial-point *classification*. To obtain separate meshed manifolds, one can cluster such points into manifolds based on their type and distance, and next use proven surface-reconstruction methods [BMR*99] from manifold point-clouds on each such cluster [KJT14a].

4.3. Garbing

Besides reconstructing the curve or surface structure of medial skeletons (Sec. 4.2), reconstructing an approximation $\tilde{\mathcal{O}}$ of the input shape \mathcal{O} from $\text{MAT}(\mathcal{O})$, or *garbing* the skeleton (Sec. 2) is another key operation in medial applications. For image (fixed-grid) skeletons, garbing is relatively easy, as it involves ‘inflating’ the skeleton voxels up to a distance equal to the maximally-inscribed ball radius. This can be done by computing the union of voxelized balls $\cup B(\mathbf{x}, r)$ of all skeleton points \mathbf{x} having a ball radius r [ASS11], or, more efficiently, computing $DT_{\text{MAT}(\mathcal{O})}$ and locally thresholding it by $r(\mathbf{x})$ [RT02]. As such, we next focus on the more complex garbing methods for analytical skeletons, which we classify in three groups: *i*) image-based methods; *ii*) object-space methods; and *iii*) m-reps and variants.

4.3.1. Image-based methods

These methods aim to produce a 2D *view* of $\tilde{\mathcal{O}}$ from $\text{MAT}(\mathcal{O})$. This can be done by *ball splatting* ([JKT13], Fig. 14): For each medial atom $a_i = (s_i, r_i)$, a 2D texture $B(a_i)$ encodes the shading and depth profile of a unit-sphere centered at the origin (Fig. 23). These 2D

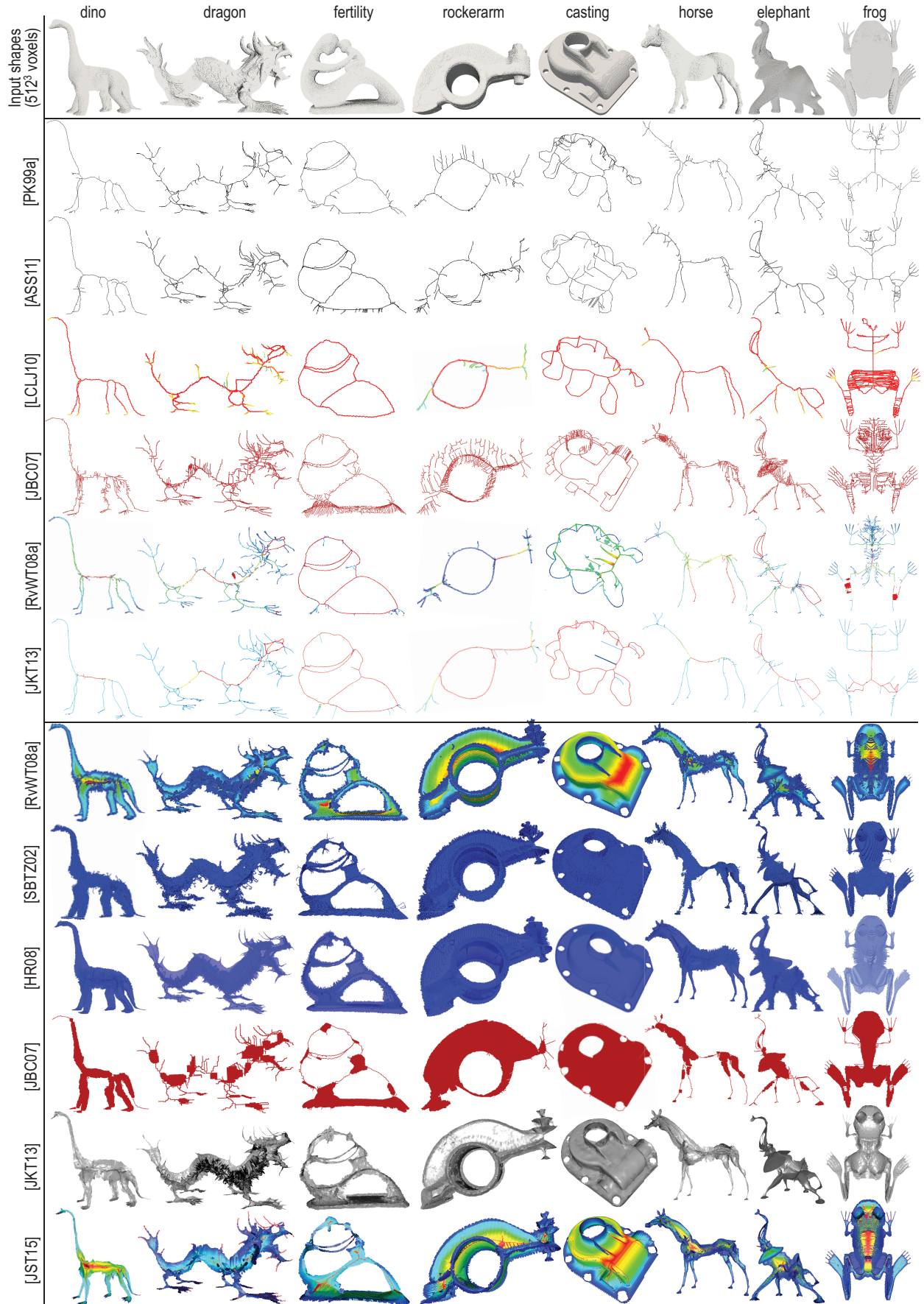


Figure 21: Qualitative comparison of six curve and six surface skeletonization methods, illustrating several differences of their outcomes.

Method		Compared with	
Type	Name	Surface skeleton	Curve skeleton
M	Huang <i>et al.</i> [HWC0*13]	–	[TZCO09]
M	Jalba <i>et al.</i> [JKT13]	[BGP10]	[RvWT08a, ATC*08]
M	Tagliasacchi <i>et al.</i> [TAOZ12]	–	[DS06, ATC*08]
M	Giesen <i>et al.</i> [BGP10]	[DZ04, BYS07, CCT09]	–
M	Au <i>et al.</i> [ATC*08]	–	[GS99, PK99b, CSYB05, DS06]
M	Dey and Sun [DS06]	–	[CSYB05]
M+V	Livesu <i>et al.</i> [LGS12]	–	[CSYB05, DS06, ATC*08, LCLJ10]
V	Liu <i>et al.</i> [LCLJ10]	–	[RvWT08a, PK99b]
V	Arcelli <i>et al.</i> [ASS11]	–	0
V	Reniers <i>et al.</i> [RvWT08a]	0	0
V	Hesselink <i>et al.</i> [HR08]	0	–
V	Ju <i>et al.</i> [JBC07]	[Ber95]	[Ber95]
V	Siddiqi <i>et al.</i> [SBTZ02]	0	–
V	Saha <i>et al.</i> [SBdB15]	0	0
V	Jalba <i>et al.</i> [JST15]	[RvWT08a, JBC07, JKT13, HR08, SBTZ02]	[JBC07, LCLJ10, ASS11, PK99b, RvWT08a]
V	Sobiecki <i>et al.</i> [SJT14]	[SBTZ02, HR08, RvWT08a, JBC07]	[SBTZ02, RvWT08a, JBC07, LCLJ10, ASS11, PK99b]
M	Sobiecki <i>et al.</i> [SYJT13]	–	[ATC*08, TZCO09, CTO*10, JT12]
M+V	Cornea <i>et al.</i> [CSM07]	–	[PK99b, GS99, CSYB05, ACK01]

Table 2: Recent 3D skeletonization comparison efforts. For each method, we show its type (Volume or Mesh), and the surface- and/or curve-skeletonization methods it is compared with. Dashes show that a method does not compute the respective (curve or surface) skeleton type. Last four rows are survey papers.

textures are translated to the points s_i , scaled with r_i respectively, and rendered parallel to the view plane, using Z-buffering. The result approximates the 3D union-of-balls $\cup B(s_i, r_i)$, i.e., the reconstruction of \mathcal{O} from its MAT, from any desired viewpoint.

While fast and simple to implement, the above ball splatting only yields correct results for orthographic projections: A screen pixel \mathbf{v} is covered by texel p , which encodes the color and depth of ball point \mathbf{x} (Fig. 23). This is equivalent to raytracing $\cup B(a_i)$ with rays parallel to the view direction. For perspective projection, a correct approach is to trace rays from each billboard pixel \mathbf{p} to the viewpoint; if a ray intersects the billboard's ball, the corresponding screen pixel \mathbf{v}' gets its color and depth from the ray-sphere intersection point \mathbf{x} . This technique is also known under the name of *impostors* [McK12]. Ball splatting methods can generate shape reconstructions from skeletons of hundreds of thousand of points at interactive rates, as their speed is bounded only by the pixel fill rate of the GPU [JKT13]. However, they formally produce a view-dependent approximation of \mathcal{O} with radial constant basis functions, so bubble-like errors become visible when zooming in or using low sampling rates for the surface skeleton (see insets in Fig. 14).

4.3.2. Object-space methods

In contrast to image-space garbing, object-space garbing creates true 3D mesh representations of the reconstructed shape. The main

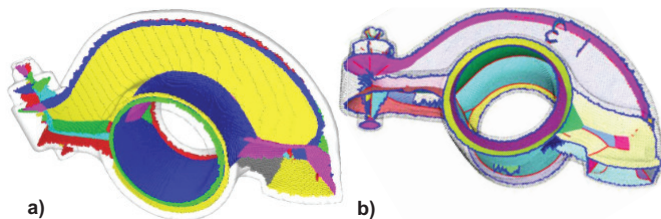


Figure 22: Surface skeleton segmentation using (a) voxel-based method in [RT08c] and (b) mesh-based method in [LK07].

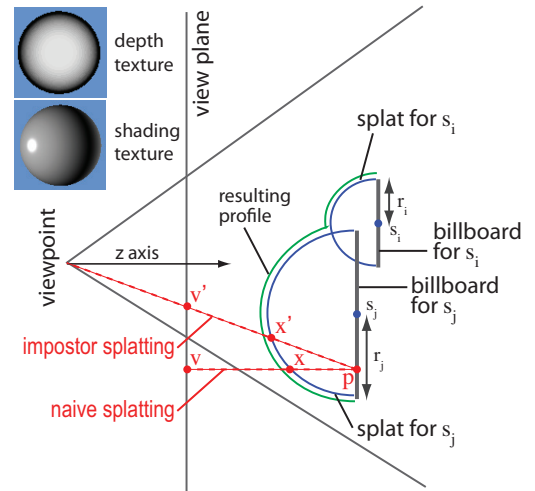


Figure 23: Ball splatting for garbing. Naive splatting works for parallel projections only [JKT13]. Impostor technique generates correct results for perspective projections [McK12].

challenge is to efficiently construct high-resolution *mesh* models of the union-of-balls surface $\cup B(\mathbf{x}, r)$. This can be done as follows.

4.3.2.1. Implicit surfaces: These methods define the surface $\cup B(\mathbf{x}_i, r_i)$ implicitly and extract it by computing the level set of the respective function [dALJ*15]. In the simplest case, the function is defined by the sum of radial distances centered at the skeleton points \mathbf{x}_i , scaled by r_i [Bli82]. This can however produce budges and undulations on $\tilde{\mathcal{O}}$ due to unwanted blending effects. Smoother reconstructions can be done by computing a convolution skeleton from the medial skeleton [ZBQC13]. For curve skeletons modeled as 1D polylines, the corresponding function can be defined by linearly interpolating the vertex radii r_i along polyline segments, which leads to reconstructions $\tilde{\mathcal{O}}$ being quad-dominant mesh with good edge flow [JLW10]. Overall, implicit surfaces are simple to implement and offer a good smoothness control of $\tilde{\mathcal{O}}$. Yet, they are in general

quite expensive, as the reconstruction quality mainly depends on the resolution of the volumetric discretization of \mathcal{O} .

4.3.2.2. Regular triangulation: Let $\{B_i = (\mathbf{x}_i, r_i)\}$ be a set of balls. The non-void intersection of two balls B_i and B_j contains two spherical caps bounded by the spheres ∂B_i and ∂B_j and by a plane orthogonal to the line $(\mathbf{x}_i, \mathbf{x}_j)$. The interior of any cap is contained in the union-of-balls $\cup B_i$. The portion of a sphere ∂B_i that does not belong to any cap represents the contribution of B_i to the surface of $\cup B_i$. This is precisely the restriction of ∂B_i to the so-called *power cell* of B_i [CKL11]. The power cell is a weighted version of a Delaunay cell obtained with the power distance metric defined by $d_{\text{pow}}(B_i, B_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 - r_i^2 - r_j^2$ and $d_{\text{pow}}(B_i, \mathbf{x} \in \mathbb{R}^3) = \|\mathbf{x}_i - \mathbf{x}\|^2 - r_i^2$. The set of power cells forms the power diagram, which is the dual of the regular triangulation of the points \mathbf{x}_i weighted by the values r_i . Hence, one of the first garbing methods relied on the regular triangulation to create a meshed representation of a skeleton's union-of-balls [AE96]. Here, the respective mesh is composed of the power cell faces that are outside any spherical caps.

4.3.2.3. Skin surfaces: A number of garbing methods are based on the so-called *skin surfaces* [Ede99], a structure widely used to model molecules. A skin surface is defined by a set of balls $\{B_i = (\mathbf{x}_i, r_i)\}$ and a shrink factor $s > 0$. Formally, the skin surface is the 3D boundary of the convex hull of the four-dimensional points $(\mathbf{x}_i, \|\mathbf{x}_i\|^2 - s^2 r_i^2)$. For $s = 1$, the skin surface is the surface of the union-of-balls $\cup B_i$. For $s < 1$, the skin surface is tangent-continuous and composed of spherical and hyperboloid patches. Several methods exist to compute triangulations of skin surfaces [CS04, KV07, CS09] which can deliver our union-of-balls surface, as noted above. However, such methods are relatively slow, *e.g.* minutes for a few hundred balls [CS04, KV07]. Computational scalability is addressed by mixed CPU-GPU implementations [CLM08], where the CPU computes the equations of the spherical and hyperboloids patches, and the GPU ray-traces these. This approach is over one order of magnitude faster, and more robust, than the CGAL [Kru15] implementation of [KV07]. However, this raytraced approach only produces a rendering of the skeleton reconstruction, rather than a 3D shape, akin to the image-based methods listed in Sec. 4.3.1.

4.3.2.4. M-Reps and variants: In contrast to modeling $\tilde{\mathcal{O}}$ as an union-of-balls with centers located on the skeleton, medial representations (*m-reps*) propose using an union of various types of surface primitives formed by the spoke vectors emerging from the sampled medial surface [PFJ*03]. Such primitives can be flexibly controlled based on the position and tangent directions at the medial samples, and lengths and angles of the spoke vectors. This allows reconstructing both the surface skeleton and the input shape by subdivision surfaces [CC78], which have guaranteed smoothness properties. A related approach is given by continuous medial representations (*cm-reps*) which use cubic B-splines and allow extracting a parametric model of $\tilde{\mathcal{O}}$ [YZG06]. While both above approaches generate a reconstruction $\tilde{\mathcal{O}}$ of higher-order continuity than the union-of-balls methods discussed earlier, they can only be applied to a single-sheet medial surface, due to parameterization constraints. To solve this issue, Pouch *et al.* remove the requirement of explicit parameterization of the medial skeleton [PTT*15]. Medial points are grouped to form separate manifold patches, based on their medial classification (Sec. 2.2.4). Next, these point-sets are triangulated and duplicated to create two separate parallel skele-

tal sheets, which are next transported to the shape surface via their spoke vectors to create the final triangulation of $\tilde{\mathcal{O}}$.

5. Applications

As noted in Sec. 1, a key driver in skeletonization research has been to use medial descriptors for a variety of shape analysis, synthesis, and processing applications. We present here a (necessarily brief) overview of the most important applications classes where surface and/or curve skeletons are an important element. For each class, we outline several representative use-cases and example applications. While skeletons are not *mandatory* tools for supporting any of the presented applications, they are, as our survey shows, useful and usable alternative tools to other methods, and as such, deserve due attention from both researchers and practitioners.

5.1. Computer animation and shape synthesis

Developing geometry-manipulation tools for end users is a core problem in geometry processing. The MAT offers an alternative to the more commonly used surface-based methods by exploiting the volumetric information of the shape it captures (Fig. 24). For instance, Storti *et al.* [STG*97] propose to use the medial domain, rather than surfaces, for shape parameterization. This lays the basis for the widely used *m-reps* of [PFJ*03] and skeleton-based shape modeling [AC02]. Intuitively, the shape can be manipulated by changing the position and radius of a few medial atoms and then reconstructing the object surface. Following a similar idea, Bloomenthal *et al.* [BL99] connected the use of *geometric* (medial) skeletons to traditional *kinematic* skeletons for shape animation, by animating the former based on the latter one. A variant hereof uses the medial domain to fit a kinematic-skeleton template and also to transfer animations between different shapes [BP07]. Traditionally, transforming vertices of animated surfaces is done by (linear) combinations of transforms defined on kinematic skeletons. While such blending weights are typically set by artists by expensive trial and error, the MAT can be used to create a convolution scheme to automatically create them, yielding natural-looking surface animations [JBPS11]. Extracting skeletons from a sequence of pre-animated meshes allows computing skinning weights, that describe how specific control points influence shape synthesis, fully automatically [SY07]. Rather than directly modifying the medial skeleton, or using a kinematic skeleton as a control structure, freeform deformation of medial surfaces can also be used to support shape deformation [YBS03, YBS07]. The volumetric nature of the MAT allows more natural large-scale deformations with local thickness preservation and correction of self-intersections. Generating kinematic skeletons and weights using medial skeletons is not restricted to surface models, but extends to volumetric data [GS01, TAM*04]. Skeletons are also used in animation to accelerate collision detection algorithms, *e.g.* by bounding volume hierarchies constructed by decomposing shapes with skeletons [LWTH01].

5.2. Geometry processing

Arguably the best-known application of curve skeletons in geometry processing is part-based segmentation of organic shapes like plants, humans, animals, and anatomic parts [XSW03, WXS06, ATC*08, SSC008]. Similarly, surface skeletons are used for patch-based segmentation of edge-rich shapes, *e.g.* man-made objects [RT08b, CLK09, KJT15]. Figure 25 (bottom row) compares six

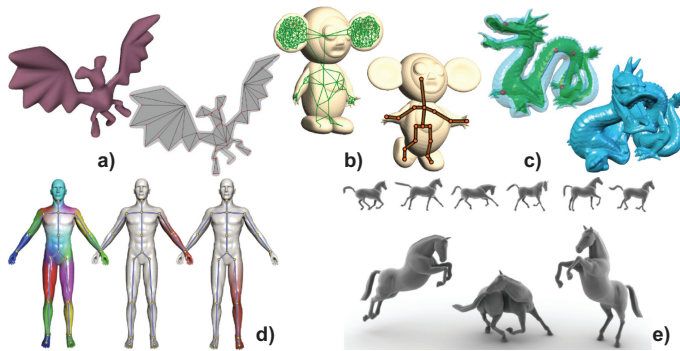


Figure 24: Animation and modeling: (a) Skeleton solid modeling [AC02]. (b) Re-using animation rigs [BP07]. (c) Volume-preserving shape deformation [YBS07]. (d) Skeleton skinning weights [JBPS11]. (e) Example-based pose extrapolation [SY07].

part-based segmentations using skeletons [LKA06, AFS06, TVD07, RT08a, FJT15] with one classical segmentation method using clustering the surface curvature [LZ04]. As illustrated, skeleton-based segmentation methods can produce comparatively good results. As skeletons compactly encode topology, they have been used to automatically correct topological problems in implicit-function iso-surfaces [WHDS04] and low-quality surface models [ZJH07]. The skeletons' approximate reconstruction power has been used for modeling organic shapes [AJC02, JLW10] as well as for correcting geometry in point-cloud data [TZCO09, CTO*10, ZST*10].

Curve skeletons compactly represent a shape as the graph of its components, their relationship and, in some cases, a rough description of their geometry [BFS00, SSGD03]. As such graphs are much smaller than corresponding mesh- or voxel-based shape representations (typically hundreds vs millions of elements respectively), they can be used to efficiently support shape comparison, matching, and retrieval, or more generally finding correspondences between shapes, via graph matching [HSKK01]. When shapes exhibit large variation in poses and surface details, graph representations obtained from (simplified) skeletons capture better the shape's overall structure than local geometric measures, *e.g.*, curvature. To better disambiguate symmetric shape parts, *e.g.*, the left vs right limbs of a body, skeletons can be stretched prior to matching [ATCO*10]. To further accelerate graph comparison, spectral decompositions of the graph's adjacency matrix can be used to match graphs in a hierarchical coarse-to-fine manner [SSDZ99, SSGD03].

Apart from the above, MAT regularization can be used for shape simplification schemes from a volumetric point of view, by eliminating small-scale shape parts corresponding to low-importance skeletal branches [TH03]. Finally, the continuous connection between medial surfaces and the shape surface [SSCO04] can be used to generate high-quality hexahedral meshes, which are essential for accurate numerical simulations in *e.g.* computational fluid dynamics [PA95, PA97, TKGC03].

5.3. Shape metrology

Skeletons can be used to measure many geometric shape properties that further support applications in shape metrology (Fig. 26). The distance-to-boundary directly delivers a way to define and estimate local shape thickness, or wall thickness [Geo10], used to estimate

e.g. the printability and mechanical resistance of 3D shapes [DK09, JKT13], anatomic tissue resistance [YP03], characterize anatomic shapes [NSK*97], and find tubular shape parts [MPS*04]. Finding similar-thickness shape parts enables high-quality shape segmentation [SSCO08, FJT15]. Separately, finding input-surface points that correspond to the boundaries of the medial surface via the feature transform allows robustly finding edges even for complex noisy shapes, which serves shape classification [RJT08, KJT15].

5.4. Medical shape analysis

Curve skeletons are well suited to describe tube-like anatomical structures, *e.g.* vessels, nerves, and elongated muscles [NdBS01, FPAB04] (see Fig. 27). This supports efficient registration of partially overlapped vessel images [AJWB03] and the flattening of their 3D structure to a plane [KWFG03]. The encoded local volumetric information can help detecting abnormalities in vascular structures, like stenosis [SHE*02] and aneurisms [SCC*04]. In virtual navigation, curve skeletons and centerlines are used to generate camera fly-through paths for the inspection of the colon [HMK*97, WDK01], lungs [PPF04], and blood vessels [BSSW99]. Alternatively, curve skeletons can be used to 'unfold' such tubular structures to create 2D flat depictions thereof, which are much easier to visualize [VWKG01]. Surface skeletons can be used to detect the subtle shallow creases separating teeth from surrounding gums for orthodontic cast segmentation [KJT14b]. Further notable applications of curve skeletons in medical image segmentation are shown in [WH10, HAM04].

6. Open challenges

Concluding our review, we have shown that 3D shape skeletonization is a very active research area, with impressive developments being shown in the few last years in terms of methods and applications. 3D skeletons have established themselves as essential tools in shape analysis and processing, much like 2D skeletons have done it in previous decades for similar tasks involving 2D shapes. However, for 3D skeletons to become tools which are as easy to use, efficient, and effective as their 2D counterparts are, several developments need to take place, as follows.

6.1. Theory

Solid techniques and tools require a solid formal background. In this respect, 3D skeletonization meets a few key challenges. First, a formal *and* universally accepted definition of curve skeletons is still due. While [DS06, RvWT08a, JKT13] do provide one, this definition still lacks a fundamental setting, similar to the classical surface-skeleton definition. Having such a definition would solve the problem of qualitative curve-skeletonization algorithm comparison, on the one hand, and focus efforts towards creating an optimal curve-skeletonization method, on the other hand.

Separately, the issue of quantitatively (and formally) comparing skeletonization algorithms is still open. As already noted, several desirable skeleton properties are still weakly measurable, *e.g.*, noise robustness, regularization, and smoothness. Providing unannounced, easily computable, formal definitions hereof is mandatory if we want to compare 3D skeletonization algorithms with explicit provisions for their potential end-users.

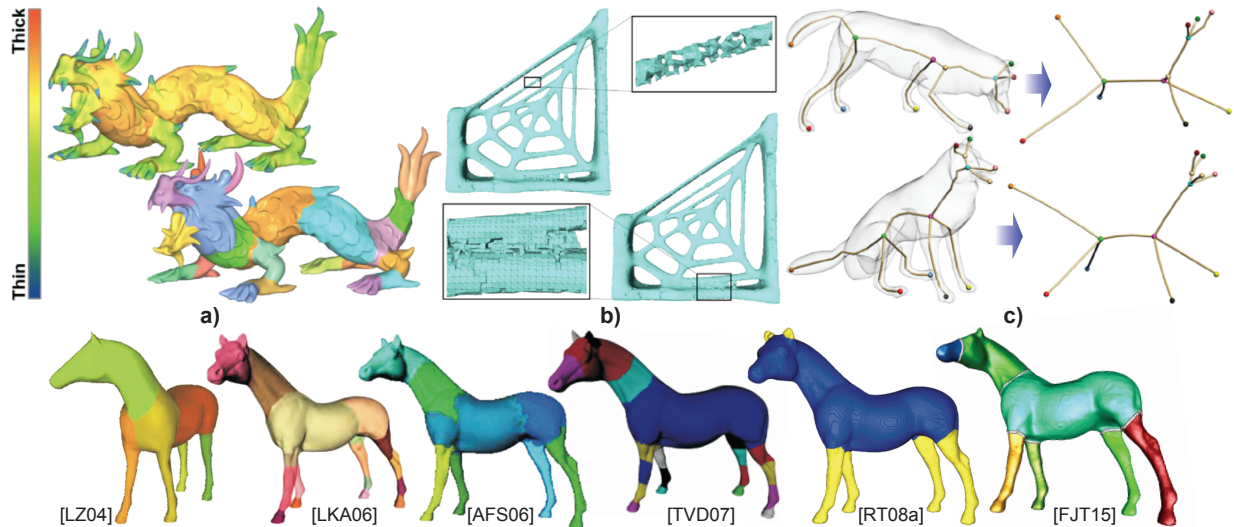


Figure 25: Geometry processing applications: (a) Shape segmentation and thickness estimation [ATC*08]. (b) Topology fixing for low-quality meshes [ZJH07]. (c) Skeleton stretching for robust pose-invariant shape matching [ATCO*10]. Bottom row: comparison of six part-based segmentation methods. Except [LZ04], all these methods use curve skeletons.

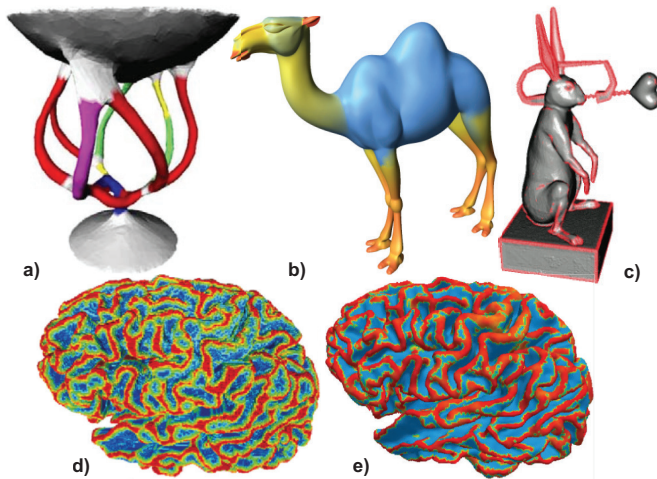


Figure 26: Metrology applications. (a) Finding tubular parts [MPS*04]. Estimating shape thickness with (b) curve skeletons [SSCO08] and (c) surface skeletons [JST15]. Finding shape edges via medial surfaces for (d) voxel shapes [RJT08] and (e) mesh shapes [KJT15].

6.2. Practice

Usable techniques require a practical setting. Here, 3D skeletonization meets many challenges. Practitioners in the field require open access to algorithm implementations, test-sets, and experience reports. 3D skeletonization is, we believe, particularly restricted here: Obtaining ready-to-use, robust, commercial-grade implementations is hard. While initiatives for disseminating algorithms and benchmarks recently appeared [Tel16], it is still very hard for application practitioners to quickly test-and-try existing methods in their specific application contexts. Albeit mundane, the need for more openness, dissemination, and comparison of 3D skeletonization algorithms is a major challenge.

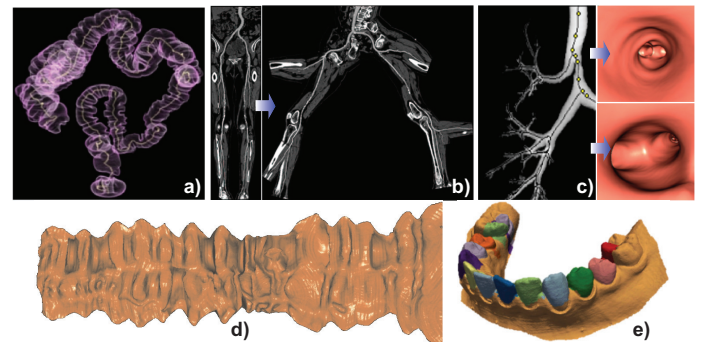


Figure 27: Medical applications: Curve skeletons (a) Virtual colonoscopy [WDK01]. (b) Vessel planar reformation [KWFG03]. (c) Virtual bronchoscopy [PFP04]. (d) Colon unfolding [VWKG01]. Surface skeletons (e) Teeth segmentation [KJT14b]

6.3. Applications

Skeletonization has emerged as a research field decades ago, in a relatively theoretical area [Blu67, Blu73]. Recent developments, especially for 3D shapes, show a great application potential for shape segmentation, simplification, denoising, and retrieval [TV08, MBC12, JKT13, SBdB15, KJT15]. Given the current developments in 3D surface-and-curve skeletonization, the basic tools are available. What still misses, is their wide deployment in many relevant applications. We believe it is the task of the research community to involve itself and show that the recent scalable, robust, and easy-to-use 3D skeletonization methods they created can indeed be effectively used to solve many concrete open shape analysis and processing problems.

7. Conclusions

We presented a survey of the state-of-the-art in computing and using three-dimensional shape skeletons. Several important observations

can be distilled from this survey. First and foremost, we see a strong and sustained interest in using 3D skeletons for a wide variety of applications, including but not limited to shape segmentation, path planning, shape metrology, animation, shape modeling, and shape matching. In all these contexts, 3D skeletons show their added-value as compared to classical boundary or volumetric representations in terms of compactly capturing, and allowing reasoning about, the shape geometry, topology, and symmetry. Originally pioneered for similar use-cases for 2D shapes, skeletons are increasingly showing their advantages in the 3D context. This trend is sustained by the advent of recent skeletonization methods which deliver accurate and high-resolution 3D skeletons with low computational and implementation efforts. In contrast to a decade ago, computing centered, detail-rich, and noise-resistant 3D skeletons for complex shapes is now a given fact. Given all above, we argue that 3D skeletons are here to stay, and will lead to increasingly more and more diverse applications. In particular, while most existing applications have focused on using the properties of the simpler to compute curve skeletons, recent developments have made surface skeletons an equally interesting, and equally practical, tool.

In the same time, the 3D skeletonization arena is complex. Tens (if not hundreds) of methods exist, each having specific assumptions and limitations. Such methods are not always easy to compare with respect of the many, often mutually conflicting, application-specific requirements. The formal analysis of intrinsic skeleton properties implied by their definitions, started by this survey, and further refined to account for properties implied by practical skeletonization methods, proves in our eyes to be a valuable practical and theoretical guide to assessing and selecting suitable methods for practical use – or finding one’s way in the wide world of 3D skeletonization.

Given the above, future developments in 3D skeletonization are numerous. On the theoretical side, formal definitions of curve skeletons, and their relation with the well-established surface skeletons, are due, and emerging. On the practical side, recent advances in efficiently and robustly computing accurate 3D surface skeletons have broken the status quo of such descriptors being complex, inefficient, and unreliable. As such, we expect a steadily increasing number of innovative applications using such descriptors in the near future.

Biographies

Andrea Tagliasacchi is an Assistant Professor of Computer Science at the University of Victoria, Canada. He received his MSc from Politecnico di Milano and a PhD from Simon Fraser University (SFU). Before joining UVic he has been a postdoctoral scholar in the Computer Graphics and Geometry Laboratory at the Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland. His research interests include computer graphics, computer vision and machine learning with a focus in geometry processing.

Thomas Delame received his computer science M. Sc. and M. Eng. degrees from ENSIMAG, France in 2010 and his Ph.D. degree in computer graphics from Burgundy University in 2014. He is now a postdoctoral researcher at Grenoble University, France. His research focus is intuitive 3D shape interaction.

Michela Spagnuolo is Research Director at CNR-IMATI. Her research interest include geometric, structural and semantic modelling

of 3D objects, with emphasis on computational topology for the analysis of shapes and for the evaluation of shape similarity. She is member of the steering committee of Shape Modeling International, EG 3D Object Retrieval, and of the EG Graphics and Cultural Heritage. She authored more than 130 reviewed papers in scientific journals and international conferences, edited two books, and is associate editor of international journals in Computer Graphics (currently, *Computers & Graphics* and *The Visual Computer*). She has been elected Eurographics Fellow in 2014.

Nina Amenta is Professor of Computer Science at the University of California at Davis. She works in computer graphics, geometric algorithms, discrete math and scientific visualization. Her work includes several algorithms for medial axis computation and the dual problem of surface reconstruction. Her biological work includes visualizing phylogeny, and algorithms for protein-protein docking. Recent work includes algorithms for hash table construction and nearest-neighbors on GPU’s. She received an Alfred P. Sloan Research Fellowship and a UC Davis Chancellor’s Fellowship. She served as the campus Director of the Center for Information Technology in the Interest of Society, and she is currently the Computer Science Department Chair at UC Davis.

Alexandru Telea received his PhD (2000) in Computer Science from the Eindhoven University of Technology, the Netherlands. Until 2007, he was assistant professor in visualization and computer graphics at the same university. Since 2007, he is professor of multiscale visual analytics at the University of Groningen, the Netherlands. His interests include 3D multiscale shape processing, scientific and information visualization, and visual analytics.

Appendix: Skeletonization Resources

Obtaining benchmark datasets for comparing 3D skeletonization methods, and implementations of such methods, is challenging. An ongoing effort in this direction is the project [Tel16] which provides an updated list of models, shape processing tools, and openly-available 2D and 3D skeletonization methods implementations for both practitioners and researchers in the field [SJT14].

Acknowledgments

This work was partially funded by the advanced grant no. 291184 EXPRESSIVE from the European Research Council (ERC-2011-ADG_20110209), and by the H2020 EC Project GRAVITATE (RE-FLECTIVE7, grant no.665155).

References

- [AB98] AMENTA N., BERN M.: Surface reconstruction by Voronoi filtering. In *Proc. ACM SCG* (1998), pp. 39–48. 10
- [AB02] AYLWARD S., BULLITT E.: Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE Trans. Med. Imag.* 21, 2 (2002), 61–75. 2
- [ABE09] ATTALI D., BOISSONNAT J. D., EDELSBRUNNER H.: Stability and computation of medial axes – a state-of-the-art report. In *Mathematical foundations of scientific visualization, computer graphics, and massive data exploration* (2009), Springer, pp. 109–125. 2, 6, 9
- [ABK98] AMENTA N., BERN M., KAMVYSSELIS M.: A new Voronoi-based surface reconstruction algorithm. *ACM TOG* (1998), 415–421. 4
- [AC02] ANGELIDIS A., CANI M. P.: Adaptive implicit modeling using subdivision curves and surfaces as skeletons. In *Proc. ACM SMA* (2002), pp. 45–52. 18, 19

- [ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *Proc. ACM SMA* (2001), pp. 249–266. [8](#), [9](#), [10](#), [11](#), [17](#)
- [AE96] AKKIRAJU N., EDELSBRUNNER H.: Triangulating the surface of a molecule. *Discr Appl Math* 71, 1–3 (1996), 5–22. [18](#)
- [AFS06] ATTENE M., FALCIDIENO B., SPAGNUOLO M.: Hierarchical mesh segmentation based on fitting primitives. *Visual Comput* 22, 3 (2006), 181–193. [19](#)
- [Ago05] AGOSTON M.: *Computer Graphics and Geometric Modelling*. Springer, 2005. [1](#)
- [AJC02] ANGELIDIS A., JEPP P., CANI M. P.: Implicit modeling with skeleton curves: Controlled blending in contact situations. In *Proc. ACM SMA* (2002), pp. 85–92. [19](#)
- [AJWB03] AYLWARD S., JOMIER J., WEEKS S., BULLITT E.: Registration and analysis of vascular images. *IJCV* 55, 2 (2003), 123–138. [19](#)
- [AKP*04] ANGUELOV D., KOLLER D., PANG H., SRINIVASAN P., THRUN S.: Recovering articulated object models from 3D range data. In *Proc. Uncertainty in Artificial Intelligence* (2004), pp. 18–26. [13](#)
- [AL01] ATTALI D., LACHAUD J.: Delaunay conforming iso-surface, skeleton extraction and noise removal. *Comp. Geom.* 19, 2 (2001), 175–189. [6](#)
- [AM97] ATTALI D., MONTANVERT A.: Computing and simplifying 2D and 3D continuous skeletons. *CVIU* 67, 3 (1997), 261–273. [10](#)
- [AMN*98] ARYA S., MOUNT D., NETANYAHU N., SILVERMAN R., WU A.: An optimal algorithm for approximate nearest neighbor searching. *J. of the ACM* 45, 6 (1998), 891–923. [11](#)
- [ASS11] ARCELLI C., SANNITI G., SERINO L.: Distance-driven skeletonization in voxel images. *IEEE TPAMI* 33, 4 (2011), 709–720. [6](#), [8](#), [9](#), [11](#), [14](#), [15](#), [17](#)
- [ATC*08] AU O. K. C., TAI C., CHU H., COHEN-OR D., LEE T.: Skeleton extraction by mesh contraction. In *Proc. ACM SIGGRAPH* (2008), pp. 441–449. [5](#), [8](#), [9](#), [12](#), [13](#), [15](#), [17](#), [18](#), [20](#)
- [ATCO*10] AU K., TAI C., COHEN-OR D., ZHENG Y., FU H.: Electors voting for fast automatic shape correspondence. *CGF* 29, 2 (2010), 645–654. [19](#), [20](#)
- [BA92] BRANDT J. W., ALGAZI V.: Continuous skeleton computation by voronoi diagram. *CVGIP: Image Understanding* 55, 3 (1992), 329–338. [10](#)
- [BBB*97] BLOOMENTHAL J., BAJAJ C., BLINN J., CANI M.-P., ROCKWOOD A., WYVILL B., WYVILL G.: *Introduction to Implicit Surfaces*. Morgan Kaufmann, 1997. [1](#)
- [BCAC04] BRUSCO N., CARMIGNATO S., ANDREETTO M., CORTELAZZO G.: Metrological analysis of a procedure for the automatic 3D modeling of dental plaster casts. In *Proc. 3DPVT* (2004), pp. 592–599. [6](#)
- [BDF*08] BIASOTTI S., DE FLORIANI L., FALCIDIENO B., FROSINI P., GIORGI D., LANDI C., PAPALEO L., SPAGNUOLO M.: Describing shapes by geometrical-topological properties of real functions. *ACM Computing Surveys* 40, 4 (2008), 1–87. [14](#)
- [Ber95] BERTRAND G.: A parallel thinning algorithm for medial surfaces. *Pattern Recogn Lett* 16, 9 (1995), 979–986. [17](#)
- [Beu94] BEUCHER S.: Digital skeletons in euclidean and geodesic spaces. *Signal Process* 38, 1 (1994), 127–141. [14](#)
- [BFS00] BIASOTTI S., FALCIDIENO B., SPAGNUOLO M.: Extended Reeb graphs for surface understanding and description. In *Proc. DGCI* (2000), Springer, pp. 185–197. [19](#)
- [BGP10] BALINT M., GIESEN J., PAULY M.: Discrete scale axis representations for 3D geometry. *ACM TOG* 29, 4 (2010), 1–10. [8](#), [9](#), [17](#)
- [BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon Mesh Processing*. A K Peters, 2010. [1](#)
- [BL99] BLOOMENTHAL J., LIM C.: Skeletal methods of shape manipulation. In *Proc. ACM SMA* (1999), pp. 44–47. [18](#)
- [Bli82] BLINN J. F.: A generalization of algebraic surface drawing. *ACM Trans. Graph.* 1, 3 (July 1982), 235–256. [17](#)
- [Blu67] BLUM H.: *A transformation for extracting new descriptors of shape*. Models for the perception of speech and visual form. MIT Press, 1967. [2](#), [3](#), [20](#)
- [Blu73] BLUM H.: Biological shape and visual science (Part I). *J Theor Biol* 38, 2 (1973), 205–287. [20](#)
- [BM94] BERTRAND G., MALANDAIN G.: A new characterization of three-dimensional simple points. *Pattern Recog Lett* 15, 2 (1994), 169–175. [11](#), [15](#)
- [BMMP03] BIASOTTI S., MARINI S., MORTARA M., PATANÉ G.: An overview on properties and efficacy of topological skeletons in shape modelling. In *Proc. IEEE SMI* (2003), IEEE, pp. 245–254. [14](#)
- [BMR*99] BERNARDINI F., MITTLEMAN J., RUSHMEIER H., SILVA C., TAUBIN G.: The ball-pivoting algorithm for surface reconstruction. *IEEE TVCG* 5, 4 (1999), 349–359. [15](#)
- [BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3D characters. *ACM TOG* 26, 3 (2007), 72(1–8). [18](#), [19](#)
- [BPS*10] BIASOTTI S., PATANÉ G., SPAGNUOLO M., FALCIDIENO B., BAREQUET G.: Shape approximation by differential properties of scalar functions. *CAG* 34, 3 (2010), 252–262. [14](#)
- [BSSW99] BARTZ D., STRASSER W., SKALEJ M., WELTE D.: Interactive exploration of extra and intracranial blood vessels. In *Proc. IEEE Visualization* (1999), pp. 389–392. [19](#)
- [BST05] BOUIX S., SIDDIQI K., TANNENBAUM A.: Flux driven automatic centerline extraction. *Med Imag Anal* 9, 3 (2005), 209–221. [8](#)
- [BTS*14] BERGER M., TAGLIASACCHI A., SEVERSKY L. M., ALLIEZ P., LEVINE J. A., SHARF A., SILVA C. T.: State of the art in surface reconstruction from point clouds. In *Eurographics 2014 STAR Reports* (2014), Eurographics. [15](#)
- [BYS07] BELYAEV A., YOSHIZAWA S., SEIDEL H. P.: Skeleton-based variational mesh deformations. *CGF* 26, 3 (2007), 255–264. [17](#)
- [CB08] COUPRIE M., BERTRAND G.: New characterizations of simple points, minimal non-simple sets and p-simple points in 2d, 3d and 4d discrete spaces. In *Proc. DGCI*. Springer, pp. 105–116. [11](#)
- [CC78] CATMULL E., CLARK J.: Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design* 10 (1978), 183–188. [18](#)
- [CC00] COSTA L., CESAR R.: *Shape analysis and classification*. CRC Press, 2000. [6](#), [8](#)
- [CCT09] CHAUSSARD J., COUPRIE M., TALBOT H.: A discrete λ -medial axis. In *Proc. DGCI* (2009), Springer, pp. 232–243. [17](#)
- [CK11] CHUANG M., KAZHDAN M.: A fast mean-curvature flow via finite-elements tracking. *CGF* (2011). [12](#)
- [CKL11] CAZALS F., KANHERE H., LORIOT S.: Computing the volume of a union of balls: A certified algorithm. *ACM Trans. Math. Softw.* 38, 1 (2011), 1–20. [18](#)
- [CKM04] CULVER T., KEYSER J., MANOCHA D.: Exact computation of the medial axis of a polyhedron. *CAGD* 21, 1 (2004), 65–98. [6](#), [11](#)
- [CL05a] CHAZAL F., LIEUTIER A.: The λ -medial axis. *Graph. Models* 67, 4 (2005). [8](#)
- [CL05b] CHAZAL F., LIEUTIER A.: Weak feature size and persistent homology: computing homology of solids in R^n from noisy data samples. In *Proc. SCG* (2005). [8](#)
- [CLK09] CHANG M., LEYMARIE F., KIMIA B.: Surface reconstruction from point clouds by transforming the medial scaffold. *CVIU*, 113 (2009), 1130–1146. [5](#), [15](#), [18](#)
- [CLM08] CHAVENT M., LEVY B., MAIGRET B.: MetaMol: High-quality visualization of molecular skin surface. *J Mol Graph Modell* 27, 2 (2008), 209–216. [18](#)
- [CS04] CHENG H.-L., SHI X.: Guaranteed quality triangulation of molecular skin surfaces. In *Proc. IEEE Visualization* (2004), pp. 481–488. [18](#)
- [CS09] CHENG H., SHI X.: Quality mesh generation for molecular skin surfaces using restricted union of balls. *Comp Geom* 42, 3 (2009), 196–206. [18](#)
- [CSM07] CORNEA N. D., SILVER D., MIN P.: Curve-skeleton properties, applications, and algorithms. *IEEE TVCG* 13, 3 (2007), 530–548. [2](#), [7](#), [8](#), [9](#), [11](#), [12](#), [17](#)
- [CSYB05] CORNEA N., SILVER D., YUAN X., BALASUBRAMANIAN R.: Computing hierarchical curve-skeletons of 3D objects. *Visual Comput.* 21, 11 (2005), 945–955. [17](#)
- [CTMT10] CAO T., TANG K., MOHAMED A., TAN T.: Parallel banding algorithm to compute exact distance transform with the GPU. In *Proc. SIGGRAPH 13D* (2010), pp. 134–141. [9](#), [11](#)

- [CTO*10] CAO J., TAGLIASACCHI A., OLSON M., ZHANG H., SU Z.: Point cloud skeletons via Laplacian based contraction. In *Proc. SMI* (2010), pp. 187–197. 12, 13, 17, 19
- [dALJ*15] DE ARAÚJO B., LOPES D. S., JEPP P., JORGE J. A., WYVILL B.: A survey on implicit surface polygonization. *ACM Comput. Surv.* (2015). 17
- [Dam06] DAMON J.: Global medial structure of regions in \mathbb{R}^3 . *Geom. Topol.* 10 (2006), 2385–2429. 5
- [DATT08] DE AGUIAR E., THEOBALT C., THRUN S., SEIDEL H.: Automatic conversion of mesh animations into skeleton-based animations. *CGF* 27, 2 (2008), 389–397. 13
- [DK09] DOUGHERTY R., KUNZELMANN K.: Computing local thickness of 3D structures with ImageJ. In *Proc. Microscopy & Microanalysis Meeting* (2009). www.optinav.com/LocalThicknessEd.pdf. 19
- [DRF12] DELAME T., ROUDET C., FAUDOT D.: From a medial surface to a mesh. *CGF* 31, 5 (Aug. 2012), 1637–1646. 9
- [DS06] DEY T., SUN J.: Defining and computing curve skeletons with medial geodesic functions. In *Proc. SGP* (2006), IEEE, pp. 143–152. 7, 8, 9, 12, 14, 17, 19
- [dvOS00] DE BERG M., VAN KREFELD M., OVERMARS M., SCHWARZKOPF O.: *Computational Geometry: Algorithms and Applications*. Springer, Berlin, 2000. 6
- [DZ04] DEY T., ZHAO W.: Approximate medial axis as a Voronoi subcomplex. *Comp. Aided Design* 36, 2 (2004), 195–202. 11, 17
- [Ede99] EDELSBRUNNER H.: Deformable smooth surface design. *Discrete and Computational Geometry* 21, 1 (1999), 87–115. 18
- [FH12] FELZENSZWALB P. F., HUTTENLOCHER D. P.: Distance transforms of sampled functions. *Theory of Computing* (2012). 3
- [FJT15] FENG C., JALBA A., TELEA A.: Part-based segmentation by skeleton cut space analysis. In *Proc. ISMM* (2015), pp. 34–46. 19
- [FLM03] FOSKEY M., LIN M., MANOCHA D.: Efficient computation of a simplified medial axis. In *Proc. SMA* (2003), pp. 135–142. 6, 8, 11
- [FPAB04] FRIDMAN Y., PIZER S., AILWARD S., BULLITT E.: Extracting branching tubular object geometry via cores. *Med Imag Anal* 8, 3 (2004), 169–176. 19
- [FSL04] FALCÃO A. X., STOLFI J., LOTUFO R. A.: The image foresting transform: Theory, algorithms, and applications. *IEEE TPAMI* 26, 1 (2004), 19–29. 6, 8, 9
- [GB85] GIBLIN P. J., BRASSETT S. A.: Local symmetry of plane curves. *Amer Math Month* (1985), 689–707. 4
- [Geo10] GEOMETRIC GLOBAL: GeomCaliper tool, 2010. geomcaliper.geometricglobal.com. 6, 19
- [GF96] GE Y., FITZPATRICK J.: On the generation of skeletons from discrete Euclidean distance maps. *IEEE TPAMI* 18 (1996), 1055–1066. 11
- [Gib00] GIBLIN P.: Symmetry sets and medial axes in two and three dimensions. In *The mathematics of surfaces IX. Proceedings of the 9th IMA conference, Cambridge, GB, September 4–7, 2000*. London: Springer, 2000, pp. 306–321. 5
- [GK04] GIBLIN P., KIMIA B.: A formal classification of 3D medial axis points and their local geometry. *IEEE TPAMI* (2004), 238–251. 5, 15
- [GMPW09] GIESEN J., MIKLOS B., PAULY M., WORMSER C.: The scale axis transform. In *Proc. SGP* (2009), pp. 106–115. 6, 8
- [GRS06] GIESEN J., RAMOS E., SADRI B.: Medial axis approximation and unstable flow complex. In *Proc. ACM SGP* (2006), pp. 327–336. 11
- [GS99] GAGVANI N., SILVER D.: Parameter-controlled volume thinning. *Graph Model Imag Proc* 61, 3 (1999), 149–164. 17
- [GS01] GAGVANI N., SILVER D.: Animating volumetric models. *Graph Model*, 63 (2001), 443–458. 18
- [HAM04] HAMARNEH G., ABUGHARBIH R., MCINERNEY T.: Medial profiles for modeling deformation and statistical analysis of shape and their use in medical image segmentation. *IJSM* 10, 2 (2004), 187–209. 19
- [Hel01] HELD M.: VRONI: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. *CGTA* 18, 2 (2001), 95–123. 6
- [HF09] HASSOUNA M., FARAG A.: Variational curve skeletons using gradient vector flow. *IEEE TPAMI* 31, 12 (2009), 2257–2274. 8, 12
- [HMK*97] HONG L., MURAKI S., KAUFMAN A., BARTZ D., HE T.: Virtual voyage: Interactive navigation in the human colon. In *Proc. ACM SIGGRAPH* (1997), pp. 27–34. 19
- [HR08] HESSELINK W., ROERDINK J.: Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. *IEEE TPAMI* 30, 12 (2008), 2204–2217. 5, 6, 7, 8, 9, 11, 17
- [HKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3D shapes. *ACM TOG* (2001), 203–212. 14, 19
- [HWC0*13] HUANG H., WU S., COHEN-OR D., GONG M., ZHANG H., LI G., CHEN B.: L1-medial skeleton of point cloud. *ACM TOG* 32, 4 (2013), 1–8. 8, 12, 17
- [JBC07] JU T., BAKER M., CHIU W.: Computing a family of skeletons of volumetric models for shape description. *Comput. Aided Design* 39, 5 (2007), 352–360. 14, 17
- [JBPS11] JACOBSON A., BARAN I., POPOVIC J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM TOG* 30, 4 (2011), 78:1–78:8. 18, 19
- [JKT13] JALBA A., KUSTRA J., TELEA A.: Surface and curve skeletonization of large 3D models on the GPU. *IEEE TPAMI* 35, 6 (2013), 1495–1508. 4, 7, 8, 9, 10, 11, 12, 15, 17, 19, 20
- [JLW10] JI Z., LIU L., WANG Y.: B-mesh: A modeling system for base meshes of 3D articulated shapes. *CGF* 29, 7 (2010), 2169–2178. 17, 19
- [JST15] JALBA A., SOBIECKI A., TELEA A.: An unified multiscale framework for planar, surface, and curve skeletonization. *IEEE TPAMI* (2015). [doi:10.1109/TPAMI.2015.2414420](https://doi.org/10.1109/TPAMI.2015.2414420). 4, 7, 8, 9, 10, 11, 17, 20
- [JT12] JALBA A., TELEA A.: Computing curve skeletons from medial surfaces of 3D shapes. In *Proc. Theoru & Practice of Computer Graphics (TPCG)* (2012), Eurographics, pp. 99–106. 8, 13, 15, 17
- [JXC*12] JIANG W., XU K., CHENG Z.-Q., MARTIN R. R., DANG G.: Curve skeleton extraction by coupled graph contraction and surface clustering. *Graphical Models* (2012). 13
- [KJT14a] KUSTRA J., JALBA A., TELEA A.: Robust segmentation of multiple intersecting manifolds from unoriented noisy point clouds. *CGF* 33, 1 (2014), 73–87. 15
- [KJT14b] KUSTRA J., JALBA A., TELEA A.: Shape segmentation using medial point clouds with applications to dental cast analysis. In *Proc. VISAPP* (2014), pp. 169–172. 19, 20
- [KJT15] KUSTRA J., JALBA A., TELEA A.: Computing refined skeletal features from medial point clouds. *Patt Recog Lett* (2015). [doi:dx.doi.org/10.1016/j.patrec.2015.05.007](https://doi.org/10.1016/j.patrec.2015.05.007). 5, 15, 18, 19, 20
- [Kru15] KRUTHOF N.: 3D skin surface meshing. In *CGAL User and Reference Manual*, 4.6.2 ed. CGAL Editorial Board, 2015. URL: <http://doc.cgal.org>. 18
- [KSKB95] KIMMEL R., SHAKED D., KIRYATI N., BRUCKSTEIN A.: Skeletonization via Distance Maps and Level Sets. *CVIU* 62, 3 (1995), 382–391. 11
- [KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *Proc. ACM SIGGRAPH* (2003), 954–961. 13
- [KTZ95] KIMIA B., TANNENBAUM A., ZUCKER S.: Shapes, shocks, and deformations I: the components of two-dimensional shape and the reaction-diffusion space. *IJCV* 15, 3 (1995), 189–224. 3
- [KV07] KRUTHOF N., VEGTER G.: Meshing skin surfaces with certified topology. *Comp Geom* 36, 3 (2007), 166–182. 18
- [KWF03] KANITSAR A., WEGENKITTL R., FLEISCHMANN D., GRÖLLER E.: Advanced curved planar reformation: Flattening of vascular structures. In *Proc. IEEE Visualization* (2003), pp. 43–50. 19, 20
- [Lan79] LANTUÉJOL C.: *La Squelettisation et son Application aux Mesures Topologiques de Mosaïques Polycristallines*. School of Mines, Paris, 1979. 14
- [LCLJ10] LIU L., CHAMBERS E., LETSCHER D., JU T.: A simple and robust thinning algorithm on cell complexes. *CGF* 29, 7 (2010), 2253–2260. 17
- [LCLTE07] LIPMAN Y., COHEN-OR D., LEVIN D., TAL-EZER H.: Parameterization-free projection for geometry reconstruction. *Proc. ACM SIGGRAPH* (2007). 12

- [Lee82] LEE D.: Medial axis transformation of a planar shape. *IEEE TPAMI*, 4 (1982), 363–369. 6, 11
- [LGS12] LIVESU M., GUGGERI F., SCATENI R.: Reconstructing the curve-skeletons of 3D shapes using the visual hull. *IEEE TVCG* 18, 11 (2012), 1891–1901. 14, 17
- [Lie03] LIEUTIER A.: Any open bounded subset of \mathbb{R}^n has the same homotopy type than its medial axis. In *Proc. ACM SMA* (2003), pp. 65–75. 4
- [LK01] LEYMARIE F., KIMIA B.: The shock scaffold for representing 3D shape. In *Visual Form*, vol. 2059. Springer, 2001, pp. 216–227. 11
- [LK03] LEYMARIE F., KIMIA B.: Computation of the shock scaffold for unorganized point clouds in 3D. In *Proc. IEEE CVPR* (2003), vol. 1, pp. 345–353. 11
- [LK07] LEYMARIE F., KIMIA B.: The medial scaffold of 3D unorganized point clouds. *IEEE TVCG* 29, 2 (2007), 313–330. 4, 5, 11, 13, 15, 17
- [LKA06] LIEN J., KEYSER J., AMATO N.: Simultaneous shape decomposition and skeletonization. In *Proc. ACM SPM* (2006), pp. 219–228. 13, 19
- [LWTH01] LI X., WOON T., TAN T., HUANG Z.: Decomposing polygon meshes for interactive applications. In *Proc. ACM I3D* (2001), pp. 35–42. 13, 18
- [LZ04] LIU R., ZHANG H.: Segmentation of 3D meshes through spectral clustering. In *Proc. Pacific Graphics* (2004), pp. 298–305. 19, 20
- [Mat83] MATHER J.: Distance from a sub-manifold in Euclidean space. In *Proc. Symp. in Pure Mathematics* (1983). 3, 5
- [MBC12] MA J., BAE S., CHOI S.: 3D medial axis point approximation using nearest neighbors and the normal field. *Visual Comput* 28, 1 (2012), 7–19. 7, 9, 11, 20
- [MBPL99] MANZANERA A., BERNARD T., PRETEUX F., LONGUET B.: Medial faces from a concise 3D thinning algorithm. In *Proc. ICCV* (1999), pp. 337–343. 14
- [MCD11] MUSUVATHY S., COHEN E., DAMON J.: Computing medial axes of generic 3D regions bounded by B-spline surfaces. *CAD* 43, 11 (2011), 1485–1495. 6
- [McK12] MCKESSON J. L.: *Learning Modern 3D Graphics Programming*. Arcsynthesis.org, 2012. 17
- [MPS*04] MORTARA M., PATANÉ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Plumber: a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies. In *Proc. ACM SMA* (2004), pp. 339–344. 19, 20
- [MQR03] MAURER C., QIU R., RAGHAVAN V.: A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE TPAMI* 25, 2 (2003), 265–270. 5, 7
- [MS96] MA C., SONKA M.: A fully parallel 3D thinning algorithm and its applications. *CVIU* 64, 3 (1996), 420–433. 14
- [MSD08] MACRINI D., SIDDIQI K., DICKINSON S.: From skeletons to bone graphs: Medial abstraction for object recognition. In *Proc. IEEE CVPR* (2008), pp. 1–8. 8
- [NdBS01] NYSTROM I., DI BAJA G., SVENSSON S.: Representing volumetric vascular structures using curve skeletons. In *Proc. ICIAP* (2001), IEEE, pp. 495–500. 19
- [NSK*97] NÄF F., SZEKELY G., KIKINIS R., SHENTON M., KÜBLER O.: 3D Voronoi skeletons and their usage for the characterization and recognition of 3D organ shape. *CVIU* 66, 2 (1997), 147–161. 19
- [NT03] NOORUDDIN F., TURK G.: Simplification and repair of polygonal models using volumetric techniques. *IEEE TVCG* 9, 2 (2003), 191–205. see also www.cs.princeton.edu/~min/binvox. 9
- [OK95] OGNIWICZ R. L., KÜBLER O.: Hierarchic Voronoi skeletons. *Patt Recog*, 28 (1995), 343–359. 6, 8, 9, 11
- [PA95] PRICE M., ARMSTRONG C.: Hexahedral mesh generation by medial surface subdivision: Part I. *Int J Numer Meth Eng* 38 (1995), 3335–3359. 19
- [PA97] PRICE M., ARMSTRONG C.: Hexahedral mesh generation by medial surface subdivision: Part II. *Int J Numer Meth Eng* 40 (1997), 111, 136. 19
- [PFJ*03] PIZER S., FLETCHER P., JOSHI S., THALL A., CHEN J., FRIDMAN Y., FRITSCH D., GASH A., GLOTZER J., JIROUTEK M., LU C., MULLER K., TRACTON G., YUSHKEVICH P., CHANEY E.: Deformable M-Reps for 3D medical image segmentation. *IJCV* 55, 2–3 (2003), 85–106. 5, 18
- [PFP04] PERCHET D., FETITA C., PRÊTEUX F.: Advanced navigation tools for virtual bronchoscopy. In *Proc. SPIE Image Processing* (2004), pp. 147–158. 19, 20
- [PH02] PROHASKA S., HEGE H.: Fast visualization of plane-like structures in voxel data. In *Proc. IEEE Visualization* (2002), pp. 29–36. 8
- [PK99a] PALÁGYI K., KUBA A.: Directional 3D thinning using 8 subiterations. In *Proc. DGCI* (1999), Springer, pp. 325–336. 14, 15
- [PK99b] PALÁGYI K., KUBA A.: A parallel 3D 12-subiteration thinning algorithm. *Graph Model Imag Proc* 61, 4 (1999), 199–221. 6, 11, 17
- [PSS*03] PIZER S., SIDDIQI K., SZEKELY G., DAMON J., ZUCKER S.: Multiscale medial loci and their properties. *IJCV* 55, 2–3 (2003), 155–179. 5, 8
- [PTT*15] POUCH A., TIAN S., TAKABE M., WANG H., YUAN J., CHEUNG A., JACKSON B., GORMAN J., GORMAN R., YUSHKEVICH P.: Segmentation of the aortic valve apparatus in 3D echocardiographic images: Deformable modeling of a branching medial structure. In *Statistical Atlases and Computational Models of the Heart*. Springer, 2015, pp. 196–203. 18
- [Pud98] PUDNEY C.: Distance-ordered homotopic thinning: A skeletonization algorithm for 3d digital images. *CVIU* 72, 3 (1998), 404–413. 11, 14
- [Ree46] REEB G.: Sur les points singuliers d’une forme de Pfaff complètement intégrable ou d’une fonction numérique. *Comptes Rendus Hebdomadaires des Séances de l’Académie des Sciences* 222 (1946), 847–849. 13, 14
- [RJT08] RENIERS D., JALBA A., TELEA A.: Robust classification and analysis of anatomical surfaces using 3D skeletons. In *Proc. VCBM* (2008), pp. 61–68. 5, 19, 20
- [RT02] RUMPF M., TELEA A.: A continuous skeletonization method based on level sets. In *Proc. VisSym* (2002), pp. 151–158. 8, 9, 11, 15
- [RT05] REMYA E., THIEL E.: Exact medial axis with Euclidean distance. *Image and Vision Computing*, 23 (2005), 167–175. 9
- [RT07] RENIERS D., TELEA A.: Tolerance-based feature transforms. In *Advances in Computer Graphics & Computer Vision* (2007), Springer, pp. 187–200. 6
- [RT08a] RENIERS D., TELEA A.: Part-type segmentation of articulated voxel-shapes using the junction rule. *CGF* 27, 7 (2008), 1837–1844. 5, 19
- [RT08b] RENIERS D., TELEA A.: Patch-type segmentation of voxel shapes using simplified surface skeletons. *CGF* 27, 7 (2008), 1954–1962. 5, 18
- [RT08c] RENIERS D., TELEA A.: Segmenting simplified surface skeletons. In *Proc. DGCI* (2008), Springer, pp. 262–274. 15, 17
- [RvWT08a] RENIERS D., VAN WIJK J. J., TELEA A.: Computing multiscale skeletons of genus 0 objects using a global importance measure. *IEEE TVCG* 14, 2 (2008), 355–368. 4, 8, 10, 12, 14, 17, 19
- [RvWT08b] RENIERS D., VAN WIJK J. J., TELEA A.: Computing multiscale skeletons of genus 0 objects using a global importance measure. *IEEE TVCG* 14, 2 (2008), 355–368. 5, 6, 11
- [RW03] RUUTH S., WETTON B.: A simple scheme for volume-preserving motion by mean curvature. *J Sci Comput* 19, 1 (2003), 373–384. 12
- [SAR95] SHEEHY D., ARMSTRONG C., ROBINSON D.: Computing the medial surface of a solid from a domain Delaunay triangulation. In *Proc. ACM SMA* (1995), pp. 201–212. 6
- [SB98] SHAKED D., BRUCKSTEIN A.: Pruning medial axes. *CVIU* 69, 2 (1998), 156–169. 8
- [SBdB15] SAHA P. K., BORGEFORS G., DI BAJA G. S.: A survey on skeletonization algorithms and their applications. *Patt Recog Lett* (2015). 2, 6, 7, 8, 11, 17, 20
- [SBT202] SIDDIQI K., BOUIX S., TANNENBAUM A., ZUCKER S.: Hamilton-Jacobi skeletons. *IJCV* 48, 3 (2002), 215–231. 2, 3, 6, 8, 9, 11, 14, 17
- [SC94] SAHA P., CHAUDHURI B.: Detection of 3D simple points for topology preserving transformations with application to thinning. *IEEE TPAMI* 16, 10 (1994), 1028–1032. 11, 15
- [SCC*04] STRAKA M., CERVENANSKY M., CRUZ A. L., KÖCHL A., SRAMEK M., GRÖLLER E., FLEISCHMANN D.: The vessel glyph: Focus & context visualization in CT-angiography. In *Proc. IEEE Visualization* (2004), pp. 385–392. 19

- [Ser82] SERRA J.: *Image Analysis and Mathematical Morphology*. Acad. Press, 1982. 14
- [SFM05] SUD A., FOSKEY M., MANOCHA D.: Homotopy-preserving medial axis simplification. In *Proc. SPM* (2005), pp. 103–110. 6, 8, 11
- [SHE*02] SORANTIN E., HALMAI C., ERDOHELYI B., PÁLAGYI K., NYÚL L., OLLÉ K., GEIGER B., LINDBICHLER F., FRIEDRICH G., KIESLER K.: Spiral CT based assessment of tracheal stenoses using 3D skeletonization. *IEEE Trans Med Imag* 21, 3 (2002), 263–273. 19
- [SJT14] SOBIECKI A., JALBA A., TELEA A.: Comparison of curve and surface skeletonization methods for voxel shapes. *Patt. Recogn. Lett.* 47 (Oct. 2014), 147–156. 2, 7, 8, 11, 14, 15, 17, 21
- [SKK91] SHINAGAWA Y., KUNII T. L., KERGOSIEN Y. L.: Surface coding based on morse theory. *IEEE CG&A* 11 (1991), 66–78. 14
- [SNT*10] SATO Y., NAKANISHI K., TANAKA H., NISHII T., SUGANO N., NAKAMURA H., OCHI T., TAMURA S.: Limits to the accuracy of 3D thickness measurement in magnetic resonance images. In *Proc. MICCAI* (2010), Springer, pp. 803–810. 6
- [SP09] SIDDIQI K., PIZER S.: *Medial Representations: Mathematics, Algorithms and Applications*. Springer, 2009. 2, 5, 8, 9, 11
- [SSCO04] SHAHAM A., SHAMIR A., COHEN-OR D.: Medial axis based solid representation. In *Proc. ACM SMA* (2004), pp. 37–44. 19
- [SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonization using the shape diameter function. *Visual Comput* 24, 4 (2008), 249–259. 18, 19, 20
- [SSDZ99] SIDDIQI K., SHOUKOFANDEH A., DICKINSON S., ZUCKER S.: Shock graphs and shape matching. *IJCV* 35, 1 (1999), 13–32. 19
- [SSGD03] SUNDAR H., SILVER D., GAGVANI N., DICKINSON S.: Skeleton based shape matching and retrieval. In *Proc. SMI* (2003), pp. 130–137. 4, 7, 19
- [ST04] STRZODKA R., TELEA A.: Generalized distance transforms and skeletons in graphics hardware. In *Proc. VisSym* (2004), pp. 221–230. 5
- [STG*97] STORTI D., TURKIYYAH G., GANTER M., LIM C., STAL D.: Skeleton-based modeling operations on solids. In *Proc. ACM SMA* (1997), pp. 141–154. 18
- [SY07] SCHAEFER S., YUKSEL C.: Example-based skeleton extraction. *CGF* (2007), 153–162. 13, 18, 19
- [SYJT13] SOBIECKI A., YASAN H., JALBA A., TELEA A.: Qualitative comparison of contraction-based curve skeletonization methods. In *Proc. ISMM* (2013), Springer, pp. 425–439. 2, 7, 8, 11, 17
- [Tag12] TAGLIASACCHI A.: Skeletal representations and applications. In *Tech. report SFU-CMPT TR 2012-55-1, School of Computing Science, Simon Fraser University* (2012). 4
- [TAM*04] THEOBALT C., AGUIAR E. D., MAGNOR M., THEISEL H., SEIDEL H.: Marker-free kinematic skeleton estimation from sequences of volume data. In *Proc. ACM VRST* (2004), pp. 57–64. 18
- [TAOZ12] TAGLIASACCHI A., ALHASHIM I., OLSON M., ZHANG H.: Mean curvature skeletons. *CGF* 31, 5 (2012), 1735–1744. 7, 9, 12, 13, 14, 17
- [Tau95] TAUBIN G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proc. ICCV* (1995), pp. 902–907. 6
- [Tel12] TELEA A.: Feature preserving smoothing of shapes using saliency skeletons. In *Visualization in Medicine and Life Sciences* (2012), Springer, pp. 153–170. 8
- [Tel16] TELEA A.: 3d skeletonization benchmark, 2016. www.cs.rug.nl/svcg/Shapes/SkelBenchmark. 20, 21
- [TF81] TSAO Y., FU K.: A parallel thinning algorithm for 3-d pictures. *CGIP* 17, 4 (1981), 315 – 331. 11
- [TH03] TAM R., HEIDRICH W.: Shape simplification based on the medial axis transform. In *Proc. IEEE Visualization* (2003), pp. 63–71. 19
- [TK01] TEK H., KIMIA B.: Boundary smoothing via symmetry transforms. *J Math Imaging Vision*, 14 (2001), 211–223. 8
- [TKGC03] TCHON K., KHACHAN M., GUIBAULT F., CAMARERO R.: Constructing anisotropic geometric metrics using octrees and skeletons. In *Proc. IMR* (2003), pp. 293–304. 19
- [TV03] TELEA A., VILANOVA A.: A robust level-set algorithm for centerline extraction. In *Proc. VisSym* (2003), pp. 185–194. 14
- [TV08] TANGELDER J., VELTKAMP R.: A survey of content based 3D shape retrieval methods. *Multimed Tools Appl* 39 (2008), 441–471. 6, 20
- [TVD07] TIERNY J., VANDEBORRE J., DAOUDI M.: Topology driven 3D mesh hierarchical segmentation. In *Proc. SMA* (2007), pp. 215–220. 19
- [TvW02] TELEA A., VAN WIJK J. J.: An Augmented Fast Marching Method for Computing Skeletons and Centerlines. In *Proc. VisSym* (2002), Springer, pp. 251–258. 6, 8, 9, 10
- [TZCO09] TAGLIASACCHI A., ZHANG H., COHEN-OR D.: Curve skeleton extraction from incomplete point cloud. In *Proc. SIGGRAPH* (2009), pp. 541–550. 13, 17, 19
- [Tzo11] TZOUMAS G.: Exact medial axis of quadratic NURBS curves. In *Proc. European Workshop on Computational Geometry* (2011). 6
- [vDvdWT06] VAN DORTMONT M., VAN DE WETERING H., TELEA A.: Skeletonization and distance transforms of 3D volumes using graphics hardware. In *Proc. DGCI* (2006), Springer, pp. 617–629. 11
- [VWKG01] VILANOVA A., WEGENKITTL R., KONIG A., GRÖLLER E.: Nonlinear virtual colon unfolding. In *Proc. IEEE Visualization* (2001), pp. 411–419. 19, 20
- [WDK01] WAN M., DACHILLE F., KAUFMAN A.: Distance-field based skeletons for virtual navigation. In *Proc. IEEE Visualization* (2001), pp. 246–253. 19, 20
- [WH10] WARD A., HAMARNEH G.: GMAT: The groupwise medial axis transform for fuzzy skeletonization and intelligent pruning. *IEEE TPAMI* 32, 6 (2010), 1084–1096. 19
- [WHDS04] WOOD Z., HOPPE H., DESBRUN M., SCHROEDER P.: Removing excess topology from isosurfaces. *ACM TOG* 23, 2 (2004), 190–208. 19
- [WL08] WANG Y., LEE T.: Curve-skeleton extraction using iterative least squares optimization. *IEEE TVCG* 14, 4 (2008), 926–936. 12, 13
- [WLK*02] WAN M., LIANG Z., KE Q., HONG L., BITTER I., KAUFMAN A.: Automatic centerline extraction for virtual colonoscopy. *IEEE Trans. Med. Imag.* 21, 12 (2002), 1450–1460. 2
- [WXS06] WERIGHI N., XIAO Y., SIEBERT J.: A functional-based segmentation of human body scans in arbitrary postures. *IEEE Trans Sys Man Cyber* 36, 1 (2006), 153–165. 18
- [XSW03] XIAO Y., SIEBERT P., WERIGHI N.: A discrete Reeb graph approach for the segmentation of human body scans. In *Proc. 3DIM* (2003), pp. 344–352. 18
- [YBS03] YOSHIZAWA S., BELYAEV A., SEIDEL H.: Free-form skeleton-driven mesh deformations. In *Proc. ACM SMA* (2003), pp. 247–253. 18
- [YBS07] YOSHIZAWA S., BELYAEV A., SEIDEL H.: Skeleton-based variational mesh deformations. *CGF* 26, 3 (2007), 255–264. 18, 19
- [YP03] YEZZI A., PRINCE J.: An Eulerian PDE approach for computing tissue thickness. *IEEE Trans Med Imag* 22, 10 (2003), 1332–1339. 6, 19
- [YZG06] YUSHKEVICH P., ZHANG H., GEE J.: Continuous medial representation for anatomical structures. *Medical Imaging, IEEE Transactions on* 25, 12 (2006), 1547–1564. 18
- [ZBQC13] ZANNI C., BERNHARDT A., QUIBLIER M., CANI M.-P.: SCALE-invariant Integral Surfaces. *CGF* 32, 8 (2013), 219–232. 17
- [Zhu99] ZHU S.: Stochastic jump-diffusion process for computing medial axes in Markov random fields. *IEEE TPAMI* (1999), 1158–1169. 13
- [ZJH07] ZHOU Q., JU T., HU S.: Topology repair of solid models using skeletons. *IEEE TVCG* 27, 3 (2007), 675–685. 19, 20
- [ZST*10] ZHENG Q., SHARF A., TAGLIASACCHI A., CHEN B., ZHANG H., SHEFFER A., COHEN-OR D.: Consensus skeleton for non-rigid space-time registration. *CGF* 29, 2 (2010), 635–644. 19